World Wide Web (2010) 13:251–274
DOI 10.1007/s11280-010-0084-2

A Wikipedia Matching Approach to Contextual Advertising

Alexander N. Pak · Chin-Wan Chung

Received: 2 July 2009 / Revised: 2 July 2009 /
Accepted: 20 January 2010 / Published online: 6 February 2010

Abstract Contextual advertising is an important part of today's Web. It provides benefits to all parties: Web site owners and an advertising platform share the revenue, advertisers receive new customers, and Web site visitors get useful reference links. The relevance of selected ads for a Web page is essential for the whole system to work. Problems such as homonymy and polysemy, low intersection of keywords and context mismatch can lead to the selection of irrelevant ads. Therefore, a simple keyword matching technique gives a poor accuracy. In this paper, we propose a method for improving the relevance of contextual ads. We propose a novel "Wikipedia matching" technique that uses Wikipedia articles as "reference points" for ads selection. We show how to combine our new method with existing solutions in order to increase the overall performance. An experimental evaluation based on a set of real ads and a set of pages from news Web sites is conducted. Test results show that our proposed method performs better than existing matching strategies and using the Wikipedia matching in combination with existing approaches provides up to 50% lift in the average precision. TREC standard measure bpref-10 also confirms the positive effect of using Wikipedia matching for the effective ads selection.

Keywords contextual advertising · wikipedia matching

A. N. Pak · C.-W. Chung (✉)
Division of Computer Science, Department of EECS,
Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea
e-mail: chungcw@kaist.edu, chungcw@cs.kaist.ac.kr

A. N. Pak
e-mail: alexpak@islab.kaist.ac.kr

1 Introduction

Internet advertising is a large and rapidly growing market nowadays. IDC[1] reported the total volume and growth rate of the worldwide, regional, and US Internet advertising spending for the fourth quarter of 2008 [8]. Worldwide spending on Internet advertising was total $65.2 billion in 2008, or nearly 10% of all ad spending across all media, and will grow 15–20% a year to reach $106.6 billion in 2011, or 13.6% of total ad spending, according to IDC's Digital Marketplace Model and Forecast. Internet advertising plays an important role in today's Internet ecosystem.

A large part of Internet advertising consists of textual advertising. Textual advertising takes mainly two forms: sponsored search and contextual advertising. Sponsored search is a placement of paid links in result pages of search engines as a response to users' queries. Contextual advertising is a type of textual advertising where ad blocks are inserted in the content of a generic Web page.

In order to improve the user experience and increase the user's attention, both the sponsored search and contextual advertising use mechanisms to select ads that are related to the user's interest. In the case of the sponsored search, users' search queries are used to select ads. Contextual advertising uses the page where ads are displayed. One of the main advantages of contextual advertising over the sponsored search is that it supports various types of Web sites, such as online magazines and personal blog pages.

The first major contextual advertising platform was provided by Google in 2003 [16]. Nowadays, all big search engines (like Yahoo! and Microsoft Live Search) provide similar services for ad publishers and Web site owners. A contextual advertising network consists of four parties:

1. *Advertiser*—usually a company that promotes their products or services, supplies ads to the network and pays for the outcome
2. *Publisher*—the owner of a Web site that places contextual ad blocks on its pages and receives payments for ad displays or clicks
3. *Ad platform*—the main system that selects ads, places them on a publisher's Web page,[2] and shares the revenue with the publisher
4. *User*—visits Web pages and interacts with ads

There exist different types of payment strategies, such as pay-per-impression (an advertiser pays for ads displays on a web-page), pay-per-click (an advertiser pays for each user's click on ads), pay-per-action (an advertiser pays for each customer that was brought by the ad) and others. One of the most used scheme in ads platforms, as well as in the research, is pay-per-click. We will assume this scheme in our paper.

Broder et al. [3] have determined the estimation of a revenue of the network, given a page p as:

$$R = \sum_{i=1..k} P(\text{click}|p, a_i)\text{price}(a_i)$$

[1] Interactive Data Corporation — www.idc.com.

[2] Usually, a Web site owner has to insert a certain JavaScript code on the page that retrieves ads from the platform.

where k is the number of ads displayed on the page p and price(a_i) is the click-price of the current ad a_i. To simplify the model and concentrate on the research issue of this topic they have ignored the pricing model and concentrated on maximizing the revenue by selecting proper ads:

$$\arg\max_i P(\text{click}\, L \mid p, a_i)$$

So, it is the interest of the ad platform to select relevant ads that increase the probability of the user's attention and as a result increase the total revenue received from the advertiser. The most difficult challenge is to find out the user's intension in order to select exactly the ads that the user wants to see. In general, such information is obtained from the content of the Web page where an advertisement is placed. The main assumption is that if an advertisement is related to the Web page content then it is relevant to the user's interest. For example, if a user is viewing a page about "traveling in Europe", then showing ads with "airplane tickets information" or "hotel information" would be probably a right choice.

A textual advertisement usually consists of a title (on the average 2–5 words), a body (5–20 words) and a link to an advertiser's Web page. In some systems, a publisher can also specify bid-phrases which are phrases that should match the page content. A typical contextual ad looks as follows:

- Title: World Wide Web
- Body: Internet and Web Information Systems Journal
- URL: http://www.springer.com/
- Bid-phrases: WWW, CS journals, computer science journals

A site owner places a block with ads in the content of the Web page. When a user opens the Web page, the ad platform should analyze the content of the page, match it to ads provided by advertisers and select ads that are most relevant to the page.

1.1 Problem statement

A traditional approach for selecting ads is based on the keyword match. A Web page is split into terms (such as words, phrases or n-grams) and they are matched against similar terms from an ad's title, body, URL or bid-phrases (if available). Certain information retrieval and natural language processing methods can be applied for this process to make it effective, such as the vector space model [15] or latent semantic indexing [5]. However, traditional keyword matching faces several problems that degrade its performance:

- *Homonyms and polysems*[3] cause semantic ambiguities and as a result lead to selecting irrelevant ads. An example of polysemy is a word "wood" which has a meaning of a piece of a tree as well as a geographical area with many trees. Homonyms can appear as common names, such as "plant" (manufacturing plant or living organism). They also appear as proper names, such as personal names (Condoleeza *Rice* or a football player Kevin *Craft*) or organizations names

[3]Homonyms are words with the same spelling but different meanings, while polysems are words that bear multiple related meanings.

Jaguar Cars, Chicago *Bulls*). The presence of such words in the page may lead to ads misplacements. For example, a placing a "grain product" advertisement on a Web page devoted to "Condoleeza *Rice*'s visit to Europe".
- *The low intersection of keywords* is caused by the limited size of the content of an ad. It is difficult to establish the text similarity for keyword matching, because keywords of pages and ads have a low intersection [14]. Also, a concept can be represented as different terms (synonyms). For example, "car" may be also referred as to "vehicle" or "automobile" and if an ad contains only one term it will not match another term contained in a page.
- *The context mismatch* occurs when an ad does not match the topic of a page, while the keyword match can be exact [3]. An example of the context mismatch could be placing ads related to "tourism in China" on a Web page about "earthquake in China".

In order to solve these problems, we propose a method that we call "Wikipedia matching". Due to the problems mentioned above, we cannot match pages and ads exactly. Therefore we introduce "reference points", to which we can relate pages and ads. Through these reference points we establish matching between pages and ads. We chose a set of Wikipedia articles evenly distributed by different topics to be the reference points. For each ad we find Wikipedia articles related to that ad. The relation is established by a text similarity measure. For a given page we follow the same procedure and find related Wikipedia articles. Using Wikipedia articles as reference points, we find ads that share same related articles as the page, and construct a ranking function. To construct the ranking function, we will explore two strategies and based on them choose a combination formula. We also show how to improve one of them using a dimension reduction technique. Finally, we show how to combine Wikipedia matching with existing solutions, which are traditional keyword matching and syntactic-semantic matching [3] to increase the relevance of selected ads. We will explore two different methods for aggregating rankings: the Borda's method and the weighted sum.

The reasons we have chosen Wikipedia among other encyclopedias and text corpora are as follows:

- Wikipedia contains wide knowledge about many different concepts, thus we can find related articles for pages and ads
- Articles in Wikipedia are regularly updated, therefore the knowledge base is always recent
- Articles contain new terms that cannot be found in other linguistic corpora (i.e. no mention of "blogging" or "Google" in British National Corpus)[4]

The probability of the negative effect from homonymy and polysemy is low for Wikipedia matching because the relevance between an article and a page (or an ad) is high due to rich contents of articles. Because we select several articles for pages and ads, even if one or few articles are not relevant, the majority of articles determine the overall matching. For example, a Web page about football player "Kevin Craft" can be matched to some article devoted to "art and crafts", however, there would be more articles on the sport and football thematic. The same reasoning is plausible for

[4]British National Corpus: http://www.natcorp.ox.ac.uk/.

Table 1 Related articles for "Windows 7 beta in January?" web-page.

Title	Score
Microsoft	0.388
Microsoft data access components	0.173
Age of empires	0.077
Search engine optimization	0.065
Mozilla firefox	0.035

the context mismatch problem. The majority of articles that are related to a page will determine overall topic of that page.

The lack of keywords problem is solved with Wikipedia matching. The probability of the intersection of keywords of pages and keywords of ads increases, because articles contain terms and definitions of the same concept in different variations.

We introduce an example to explain how Wikipedia matching works. For instance, we have a page "Windows 7 beta in January?" about the release of a beta version of the Microsoft Windows operating system. Keyword matching may give several candidates of ads to place in the Web page:

1. "Windows Live for Mobile"
2. "Windows and Doors"

As we can see, both ads are matched by the keyword "window". However in the second case, this matching would lead to placing of an irrelevant ad, because we assume that the user is interested in software and computers rather than in the home repair. According to the Wikipedia matching algorithm, our method first determines reference points — Wikipedia articles that are most related to the page. Table 1 displays titles of articles, used in this example, that are similar to the web-page. The column "Score" indicates the similarity score between an article and the page calculated by the cosine measure (details will be presented in Section 3). Next, we find related articles for the ads (Tables 2 and 3). Because the ad for "Windows Live" has common "Microsoft" article with the page, it will be correctly chosen for the placement, while the "Windows and Doors" ad will be not.

Because the ad for "Windows Live" has common "Microsoft" article with the page, it will be correctly chosen for the placement, while the "Windows and Doors" ad will not be.

1.2 Contributions

The contributions of the paper are as follows:

– *Wikipedia matching algorithm*: We propose an efficient method for selecting relevant ads for a given page. We use Wikipedia articles as reference points to calculate a similarity score between pages and ads. Wikipedia matching can

Table 2 Related articles for "Windows Live for Mobile".

Title	Score
Opera (web browser)	0.064
Microsoft	0.023
Lost (TV series)	0.013
Latter days	0.013

Table 3 Related articles for "Windows and Doors".

Title	Score
2012 Summer olympics bids	0.119
Scottish parliament building	0.025
Construction of the WTC	0.021
Providence, Rhode Island	0.015

solve problems of traditional approaches to ads matching. Those problems are homonymy and polysemy, the low intersection of keywords, and the context mismatch.
- *Use of the Wikipedia similarity score with existing solutions*: We show how our method can improve the accuracy of existing matching strategies, which are keyword matching and semantic-syntactic matching. We show that, by using our proposed similarity score, their performance is significantly improved.
- *Evaluation experiments*: We show that our proposed method improves existing approaches by increasing the average precision of selected ads. We also evaluate our method using TREC standard measure bpref-10 which confirms the positive effect of using Wikipedia matching.

1.3 Organizations

The rest of the paper is organized as follows. In Section 2, we discuss prior works on contextual advertising. In Section 3, we explain our proposed Wikipedia matching method. We show how to combine ads ranking by Wikipedia matching with the existing techniques to improve their performance in Section 4. Our experimental setup and evaluation results are presented in Section 5. Finally, we conclude our work in Section 6.

2 Related work

2.1 Keyword matching

The study on contextual advertising is emerging with the growth of the Internet advertising market. The simplest and most straightforward method for selecting ads is to choose ads based on the text similarity [10]. In order to calculate a text similarity between a web-page and an ad, the cosine measure is used [15]. The cosine measure is a technique often used in information retrieval for calculating similarities between text documents or a text document and a search query.

One of the first results on the research about contextual advertising was presented by Ribeiro-Neto et al. in [14] where the vector space model for representing pages and ads was used. In their work, authors addressed the problem of low intersection between vocabularies of pages and ads. They have called this problem — the vocabulary impedance. In order to solve the problem Ribeiro-Neto et al. suggested to augment a page with additional keywords taken from other web-pages that are similar to the considered page. They have called this approach as impedance coupling strategy. The authors have explored ten different strategies for matching ads, which

use different parts of pages and those of ads. First five strategies use the proposed impedance coupling strategy and the latter five do not.

The winning strategy uses impedance coupling and is based on matching an ad using its keywords to a page. Evaluation tests show 60% improvement of the average precision when using the winning strategy compared with the last five strategies.

Murdock et al. apply a noisy-channel approach in [11] representing the problem as the sparseness of the advertisement language. The authors assume that an ad can be seen as a noisy translation of a page. Using this assumption, authors select ads that provide the best translation for a given page. To obtain a relevance score, the authors use algorithms used in machine translation that determine the quality of machine translated texts. Those are: NIST and BLEU [18].

Evaluation experiments of [11] showed that the use of the proposed machine translation features improves the performance over the baseline system, which is based on cosine similarity features.

However the main drawback of approaches based on keyword matching is due to the problems we have mentioned before (i.e. homonymy, polysemy and context mismatch) can dramatically degrade the relevance of selected ads.

2.2 Semantic advertising

Semantic advertising applies semantic technologies to online advertising solutions. This technology semantically analyzes every web page in order to properly understand and classify the meaning of a web page, and accordingly ensures that the web page contains the most appropriate advertising. Semantic advertising increases the chance that the viewer will click-thru because only advertising relevant to what they are viewing, and therefore their interests, should be displayed.

A recent research by Broder et al. [3] proposed a semantic approach to contextual advertising. The authors address problems of ambiguous keywords (homonymy and polysemy) and ambiguous page context (context mismatch). In order to overcome those problems, authors have proposed to apply automatic classification for pages and ads. The obtained classification information helps to filter out irrelevant ads and therefore increase the performance of ads selection. The authors use a commercial ontology, built especially for advertising purposes, to classify pages and ads. The ontology represents a hierarchical structure of advertising queries, and contains around 6,000 nodes. A hierarchical SVM, a log-regression classifier and the Rocchio's framework were tested for the document classification. The Rocchio's classifier provided the best results. Classified pages are then matched to ads by calculating the topical distance which is referred to as the semantic similarity score. The paper also proposes the semantic-syntactic matching, which combines the proposed semantic approach with traditional keyword matching to achieve higher results.

While the proposed method showed good results, the method, however, is sensitive to the classification precision. The obtained classification precision in the paper is 70% for pages and 86% for ads, which gives around 60% probability (70% multiplied by 86%) of a chance for a successful semantic match.

In the follow-up work to [3], Anagnostopoulos et al. [1] investigated the issue of the network latency and the system load. A technique for ads 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 allows lowering the

network traffic between a Web page and an ad platform along with decreasing the system load while sacrificing only 1%–3% of ads relevance. However, the problem of the classification precision still remains for this approach.

We have implemented the method of semantic matching approach proposed by Broder et al. in [3] and we will use it as a baseline along with the traditional keyword matching. In their work, authors use a commercial taxonomy for classifying pages and ads. The taxonomy is a property of Yahoo! Corp. and it is not available publicly. Therefore, we use a taxonomy from the OpenDirectory project (ODP).[5] We choose OpenDirectory because it is open source and it is regularly maintained by more than 82,000 editors.

3 Wikipedia matching

In this section, we describe our proposed method for matching ads avoiding the problems of traditional approaches. Our method uses similarity to Wikipedia articles as an additional feature. The overall scheme of our approach is as follows. First, we find similar Wikipedia articles for the given page using the cosine similarity. Next we find similar articles for each ad using the cosine similarity. Once we obtain similar articles for the considered page and all ads, we calculate an overall matching score. Based on that score we rank ads such that the ad with the highest rank is considered to be the most relevant one for the page.

We explore two different ways to construct the ranking function. The first method assumes Wikipedia articles to be labels (or categories) and uses the dot product to calculate the final measure that we call *Wikipedia similarity*. The second method assumes Wikipedia articles to represent coordinates in a multidimensional space and uses the Euclidean distance to calculate the overall score. We call it *Wikipedia distance*. In this method, the ranking is done according to the obtained Wikipedia distance: ads with smaller distances to the page are ranked higher. We also apply a dimension reduction technique to improve the performance of the latter method. In the next section, we show how to combine our ranking functions with keyword matching and semantic-syntactic matching to increase the precision of those methods.

3.1 Keyword extraction

Before we explain the proposed method, we will describe how the dataset is prepared. To implement the method, we need to prepare a set of web-pages (where ads will be displayed), a set of ads, and a set of Wikipedia articles to be used in our method.

We have downloaded a set of 100 news pages that were linked from Google News portal.[6] We have selected Web pages evenly among available categories (Top Stories, Business, Electronics, Sci/Tech, Entertainment, Health, Most Popular).

[5] http://dmoz.org

[6] http://news.google.com

Before storing pages in the database, each of them was processed with a content extraction tool [9]. It extracts the main content by analyzing an HTML DOM tree, and prunes unnecessary parts such as navigation links and decoration elements. Thus, out of the whole page we get only its title and the main content, which is a news story.

Next, we extract words from the title and the content of a page, and remove stopwords (i.e. words that bear no meaning, such as articles, prepositions etc.). We use a list of stopwords from the Snowball project.[7]

Each word is then processed with a stemming algorithm which truncates suffixes of the word, and reduces it to a stem [12, 13]. We experimented with the stemming algorithm and a lemmatization process.[8] The advantage of the lemmatization is the ability to capture different forms of a word (such as "better" and "good") while a stemming algorithm cannot do that. However, the lemmatization algorithm is more complex and it fails for words that are not in the dictionary. Moreover, the lemmatization also requires a part of speech information which can be ambiguous. For example, word "saw" can be a noun "saw" or a verb "to see" in a past tense, and we need to know whether it is a noun or a verb for the correct lemmatization. Therefore we decided to use stemming, as it runs faster and can be applied to any word.

Finally, we form n-grams out of stemmed words. We use unigrams for Wikipedia matching and keyword matching, and we use bigrams in the text classification for semantic matching.

To form a dataset of ads, we queried search engines with simple queries such as "education" and "computers" that were formed out of titles of categories from the OpenDirectory project. For a given query, we receive a result page with blocks of sponsored search ads. Thus, we collected 7,996 ads that form our ads dataset. Next, ads are put through the same process as pages, i.e. tokenization, stemming and stopwords filtering.

For Wikipedia matching, we selected 1,000 featured articles[9] from Wikipedia. Featured articles are presented in 29 different topics (Art, Business, History and etc.). We selected articles across all the topics, such that each topic is fairly represented with articles. In our experiments, we used different articles, however, they did not affect significantly the results as long as all the topic were evenly covered. We have experimented with various number of articles and 1,000 gave us the best results (more details in Section 5.7). In addition, using more articles increases the computational cost.

Because Wikipedia pages have the same HTML structure, it was easy to select only the main content avoiding unnecessary elements. The process of extracting terms from articles is similar to those from pages and ads.

[7] http://snowball.tartarus.org/

[8] Process of finding the lemma of a word, i.e. its initial form.

[9] http://en.wikipedia.org/wiki/Wikipedia:Featured_articles

3.2 Finding similar articles

To find similar articles for the given page p and each of ad a_i we proceed as follows. pk is a set of terms (keywords) extracted from the content (title, headings, text) of page p:

$$pk = \{pk_1, pk_2, ..., pk_q\} \tag{1}$$

ak_i is a set of terms extracted from the title and the body of an ad a_i:

$$ak_i = \{ak_1^i, ak_2^i, ..., ak_m^i\} \tag{2}$$

We select reference points — a set of Wikipedia articles: $w = \{w_1, w_2, ..., w_s\}$. Similarly, wk_i is the set of terms extracted from the title and the body of an article w_i:

$$wk_i = \{wk_1^i, wk_2^i, ..., wk_n^i\} \tag{3}$$

For each term of a page we calculate its *tf-idf* value. The *tf* part is the number of occurrences of the term in the page. For the *idf* part we calculate the number of page in which this term occurs:

$$\text{idf}(pk_i) = \log \frac{N(\text{pages})}{C(pk_i, \text{pages})}$$

where $N(\text{pages})$ is the total number of pages in the dataset and $C(pk_i, \text{pages})$ is the number of pages containing term pk_i. Finally, the formula for calculating *tf-idf* of a term pk_i is as follows:

$$\text{tfidf}(pk_i) = C(pk_i) \cdot \log \frac{N(\text{pages})}{C(pk_i, \text{pages})} \tag{4}$$

Similarly, we calculate *tf-idf* values for ads terms and article terms as follows.

$$\text{tfidf}(ak_j^h) = C(ak_j^h) \cdot \log \frac{N(\text{ads})}{C(ak_j^h, \text{ads})} \tag{5}$$

$$\text{tfidf}(wk_j^i) = C(wk_j^i) \cdot \log \frac{N(\text{articles})}{C(w_j^i, \text{articles})} \tag{6}$$

$\mathbf{p}$ is the vector of *tf-idf* values of the terms of page p. Similarly, we define a vector $\mathbf{w_i}$ for article w_i and vector $\mathbf{a_h}$ for ad a_h. Next, we calculate the cosine similarity between the page and each article. The cosine similarity is equal to the cosine value of the angle between the vector of the page and the vector of the article [10]:

$$sim(p, w_i) = \cos \angle \mathbf{p}, \mathbf{w_i} = \frac{\mathbf{p} \cdot \mathbf{w_i}}{|\mathbf{p}| \cdot |\mathbf{w_i}|}$$

$$sim(p, w_i) = \frac{\sum_{\forall j, pk_j = wk_j^i} \text{tfidf}(pk_j) \cdot \text{tfidf}(wk_j^i)}{\sqrt{\sum_{j=1}^{q} \text{tfidf}(pk_j)^2} \cdot \sqrt{\sum_{j=1}^{n} \text{tfidf}(wk_j^i)^2}} \tag{7}$$

We rank articles according to the similarity score in a decreasing order and select a set $\{pw\}$ from the top-N of articles (in our experiments we use $N = 100$).

Similarly, we calculate the cosine measure for ads:

$$sim(a_h, w_i) = \frac{\sum_{\forall j, ak_j^h = wk_j^i} \text{tfidf}\left(ak_j^h\right) \cdot \text{tfidf}\left(wk_j^i\right)}{\sqrt{\sum_{j=1}^{q} \text{tfidf}\left(ak_j^h\right)^2} \cdot \sqrt{\sum_{j=1}^{n} \text{tfidf}\left(wk_j^i\right)^2}} \tag{8}$$

For each ad, we select the top-N of the most similar articles: $\{a_h w\}$.

When we obtain a set of articles for the page and a set of articles for each ads, there are two strategies we propose to calculate overall similarity score between the page and each ad.

3.3 Wikipedia similarity

We can consider Wikipedia articles to be document categories. Indeed, if a document is similar to a Wikipedia article, this article can be seen as a category for the document. Therefore, the first strategy assumes that top-N similar articles are labels (or categories) for the considered page and the cosine measure is a probability value. Thus, we can construct a vector of cosine similarity values for the page:

$$\left(\text{sim}(p, pw_1), sim(p, pw_2), \ldots, \text{sim}(p, pw_N)\right)$$

The same way for each ad, we construct a vector of cosine similarity values:

$$\left(\text{sim}(a_h, a_h w_1), \text{sim}(a_h, a_h w_2), \ldots, \text{sim}(a_h, a_h w_N)\right)$$

A natural way to calculate similarity between the page and an ad would be to compute a dot product of their vectors:

$$\text{wsim}(p, a_h) = \sum_{\forall j, pw_j = a_h w_j} \text{sim}(p, pw_j) \cdot \text{sim}(a_h, a_h w_j) \tag{9}$$

The obtained value is the Wikipedia similarity score that we can use for selecting ads for a given page. So the ad with the highest Wikipedia similarity score should be considered as the most relevant for the given page.

3.4 Wikipedia distance

Another way to look at Wikipedia articles that are similar to a document is to imagine a multidimensional space where each dimension is represented by a Wikipedia article. The similarity score between an article and the document is the coordinate of the document in the dimension of the article. According to this model, similar documents should be located close by because they are similar to the same set of articles and therefore have similar coordinate values for the corresponding dimensions. Then, a natural way to calculate a similarity between two documents is to compute the Euclidean distance according to their coordinates.

Let PW be a point representing the page p with the following coordinates:

$$PW = \left(\text{sim}(p, pw_1), \text{sim}(p, pw_2), \ldots, \text{sim}(p, pw_N)\right)$$

Let $A_h W$ be a point representing an ad a_h with the following coordinates:

$$A_h W = \left(\text{sim}(a_h, a_h w_1), \text{sim}(a_h, a_h w_2), \ldots, \text{sim}(a_h, a_h w_N)\right)$$

The distance between these two points is calculated as:

$$\mathrm{wdist}(p, a_h) = \sqrt{\sum_{\forall j,\, pw_j = a_h w_j} \left(\mathrm{sim}(p, pw_j) - \mathrm{sim}(a_h, a_h w_j)\right)^2} \tag{10}$$

We call the obtained value Wikipedia distance, and the ad with lowest value of Wikipedia distance would be considered as the most relevant for the given page.

3.5 Dimension reduction

Because we use only the top-N of the most similar articles as coordinate values for each page and each ad to calculate the Wikipedia distance, other coordinates are regarded as equal to zero. Indeed, their value is very close to zero, because the similarity is very low or absent, thus our assumption of disregarding those coordinate does not affect much the performance. Therefore if we represent all pages and ads in a matrix with rows corresponding to points and columns corresponding to the coordinates of these points in each dimension, then such matrix would have the majority of values equal to zero. If we use top-100 articles and the total number of articles is 1,000, then 90% of matrix values will be filled with zeros. This leads to an idea of reducing the number of dimensions, while keeping the important data.

A popular technique for dimension reduction is principal component analysis (PCA). It has been shown recently [6] that PCA automatically projects to the subspace where the global solution of K-means clustering lie, and thus facilitates K-means clustering to find near-optimal solutions. This means by applying PCA to the set of points representing pages and ads, we automatically group together similar pages and ads. After we have performed PCA and obtained a set of coordinates in a reduced dimension space, we follow the same procedure as before to calculate Wikipedia distance.

In order to apply PCA, first we construct a matrix containing coordinates of all pages and ads:

$$M = \begin{bmatrix} \mathrm{sim}(p_1, w_1) & \cdots & \mathrm{sim}(p_1, w_R) \\ \vdots & \ddots & \vdots \\ \mathrm{sim}(p_Q, w_1) & \cdots & \mathrm{sim}(p_Q, w_R) \\ \mathrm{sim}(a_1, w_1) & \cdots & \mathrm{sim}(a_1, w_R) \\ \vdots & \ddots & \vdots \\ \mathrm{sim}(a_T, w_1) & \cdots & \mathrm{sim}(a_T, w_R) \end{bmatrix}$$

where Q is the total number of pages, T—the total number of ads, and R—the total number of articles. Next, we compute a covariance matrix S of M and find an eigenvector for S. We take the first eigenvector and use it to perform a dimension transformation (reduction).

Finally, we obtain matrix M^* containing coordinates representing pages and ads in a reduced space. In our research we use number of dimensions for the reduced space 100. When we obtain matrix M^*, we calculate the Wikipedia distance between a given page and an ad in the reduced space. The formula is the same as that for

the original space (10), and the ad with the lowest value of Wikipedia distance is considered as the most relevant for the given page.

4 Combining ranking functions

In this section, we show how to combine our proposed ranking functions with existing solutions. We will consider combinations of our proposed ranking methods (Wikipedia similarity, and Wikipedia distance in original and reduced dimensions) with keyword matching and semantic-syntactic matching. We will show that using Wikipedia matching will improve the performance of those methods.

4.1 Weighted sum

One of the popular techniques to combine different ranking methods is to use the weighted sum. In the machine learning area, such technique is also known as voting. We consider different ranking methods as "experts" which give votes about the similarity of page-ad pair. Then, we assign a weight (an importance) for each expert and calculate a weighted sum. The computed value is a combined value from different methods. Broder et al. in [3] use weighted sum to compute semantic-syntactic score. If we have a set of L ranking functions $\{r_j\}$ then the overall score is calculated as:

$$\text{score} = \sum_{j=1}^{L} w_j r_j$$

where w_j is a weight for a function r_j and necessary conditions are: $w_j \geq 0, \forall j$ and $\sum_{j=1}^{L} w_j = 1$

The advantage of the weighted sum is its simplicity and the ability to control the performance of the combined method by tuning weights. However, tuning parameters is always a very delicate and time consuming process. It also usually involves a human expert to control the process. The system parameters (weights) should be updated regularly, in order to give the best performance, because the environment can also be regularly changed. If the number of different ranking methods increases, the time needed for tuning parameters increases exponentially.

Another issue to deal with is the normalization. Different matching functions may produce values of a different range and scale. In order to combine them, we need to normalize their values. A usual way is to obtain a value between 0 and 1, where 1 indicates strong matching and 0 indicates no matching. However, it is not always easy to determine a good normalization method for a function. We show how to normalize values of Wikipedia similarity and Wikipedia distance so we can combine them with other methods using the weighted sum.

Theoretically, the Wikipedia similarity function can take a value between 0 and N inclusively. But due to small values of similarity scores between articles and pages/ads, in practice, if we use $N = 100$, we obtain values between 0 and 1.11. The obtained average of maximum values is $ave = 0.11$:

$$ave = \frac{1}{|P|} \sum_{\forall p \in P} \max_{\forall a_h \in A} \text{wsim}(p, a_h)$$

where P is the set of pages and A set of ads. We normalize the similarity score in order to get values in the interval [0, 1], as follows:

$$\text{wsim}'(p, a_h) = \begin{cases} 1 & \text{if wsim}(p, a_h) \geq ave \\ \dfrac{\text{wsim}(p, a_h)}{\text{ave}} & \text{otherwise} \end{cases} \tag{11}$$

We use the average of maximum values rather then the maximum value, because we do not want to discriminate small values.

The Wikipedia distance takes the minimum value of 0, which indicates the closest possible matching, and its value grows if there is less similarity between a page and an ad. To normalize the Wikipedia distance value, we find the minimum and the maximum values in the whole dataset and then use them for the normalization:

$$\text{wdist}'(p, a_h) = (\max - \text{wdist}(p, a_h))/(\max - \min) \tag{12}$$

For our dataset, we have obtained values of $\max = 1.8$ and $\min = 0.03$ for the reduced dimensions and $\max = 1.9$ and $\min = 0.16$ for the original space.

In this paper, we consider the following combinations by the weighted sum (all the weights were determined experimentally and the experimental results are shown below):

- Semantic-syntactic matching by Broder et al.:

$$\text{KS}(p, a_h) = \alpha \cdot \text{ksim}(p, a_h) + (1 - \alpha) \cdot \text{semsim}(p, a_h)$$

  ($\alpha = 0.8$)
- Keyword matching and normalized Wikipedia similarity:

$$\text{KW}(p, a_h) = \alpha \cdot \text{ksim}(p, a_h) + (1 - \alpha) \cdot \text{wsim'}(p, a_h)$$

  ($\alpha = 0.4$)
- Keyword matching and normalized Wikipedia distance:

$$\text{KE1}(p, a_h) = \alpha \cdot \text{ksim}(p, a_h) + (1 - \alpha) \cdot \text{wdist'}(p, a_h)$$

  ($\alpha = 0.1$)
- Keyword matching and normalized Wikipedia distance in the reduced dimension space:

$$\text{KE2}(p, a_h) = \alpha \cdot \text{ksim}(p, a_h) + (1 - \alpha) \cdot \text{wrdist'}(p, a_h)$$

  ($\alpha = 0.1$)
- All the considered methods: keyword matching, semantic matching, normalized Wikipedia similarity, normalized Wikipedia distance, and normalized Wikipedia distance in the reduced dimension space:

$$\text{ALL}(p, a_h) = \sum_i \alpha_i \cdot \text{fsim}_i(p, a_h)$$

  where fsim = {ksim, semsim, wsim', wdist', wrdist'} and weights are: $\alpha = \{0.05, 0.03, 0.07, 0.45, 0.4\}$

In order to determine weights, different optimization algorithms can be used. In our paper, we use a simple iterative approach to find a set of weights that provides the

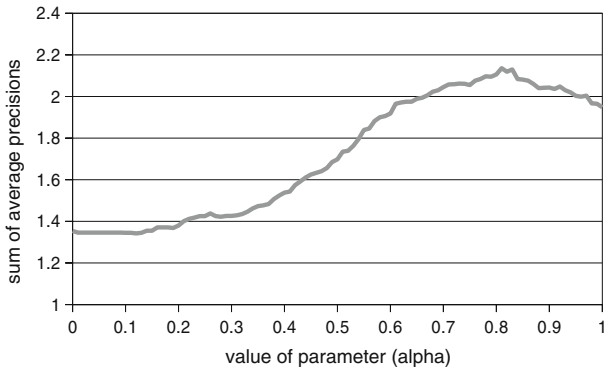

Figure 1 Determining parameter value for syntactic-semantic matching.

maximum value, i.e. we change the value of the parameters from 0 to 1 with a small step and calculate a sum of average precisions of selected ads. A parameter's value that provides the maximum for a sum of average precisions is finally selected. On Figures 1, 2, 3 and 4, the sums of average precisions of selected ads over the values of the parameter α are presented. We cannot visualize the results for parameter estimation of the ALL combination, but the approach is the same as the other combinations.

4.2 Borda's count

To overcome the problem of tuning system parameters and the normalization, we propose to use the Borda's method for aggregating ranks [7]. The Borda's method is a single-winner election method in which voters rank candidates in the order of preference. The Borda's method determines the winner of an election by giving each candidate a certain number of points corresponding to the position in which he or she is ranked by each voter. Once all votes have been counted, the candidate with the most points is the winner.

The computation of the Borda's score is very easy and can be done in linear time. The number of computations grows linearly to the number of ranking methods.

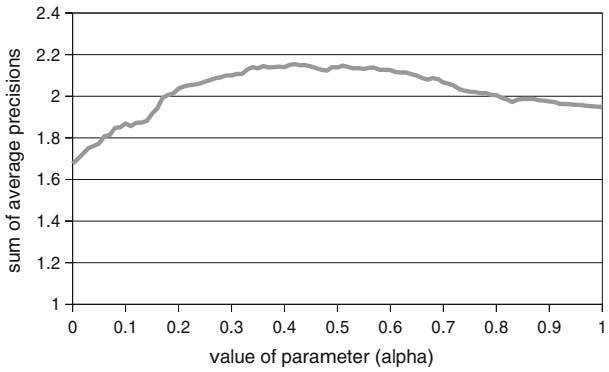

Figure 2 Determining parameter value for the combination of keyword matching and Wikipedia similarity.

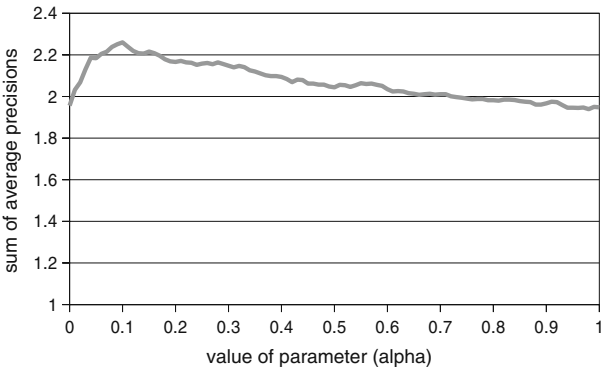

Figure 3 Determining parameter value for the combination of keyword matching and Wikipedia distance.

Therefore, we can easily compute a combination of different ranking methods without a need to tune system parameters every time.

In this paper, we consider the following combinations by the Borda's count:

– Keyword matching and Wikipedia similarity:

$$\mathrm{BKW}(p, a_h) = \mathrm{Borda}(\mathrm{ksim}(p, a_h), \mathrm{wsim}(p, a_h))$$

– Keyword matching and Wikipedia distance:

$$\mathrm{BKE1}(p, a_h) = \mathrm{Borda}(\mathrm{ksim}(p, a_h), \mathrm{wdist}(p, a_h))$$

– Keyword matching and Wikipedia distance in the reduced dimension space:

$$\mathrm{BKE2}(p, a_h) = \mathrm{Borda}(\mathrm{ksim}(p, a_h), \mathrm{wrdist}(p, a_h))$$

– All the considered methods: keyword matching, semantic matching, Wikipedia similarity, Wikipedia distance, and Wikipedia distance in the reduced dimension space:

$$\mathrm{BALL}(p, a_h) = \mathrm{Borda}(\mathrm{ksim}(p, a_h), \mathrm{semsim}(p, a_h), \mathrm{wsim}(p, a_h), \mathrm{wdist}(p, a_h), \mathrm{wrdist}(p, a_h))$$

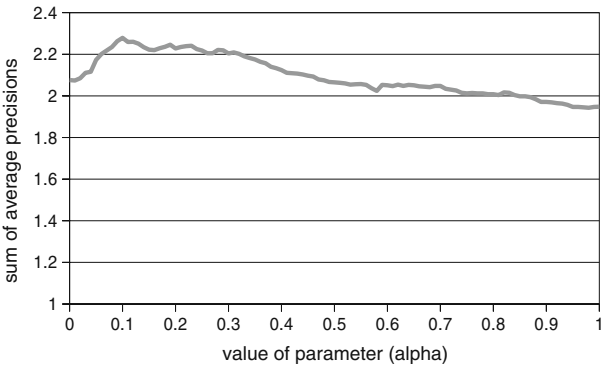

Figure 4 Determining parameter value for the combination of keyword matching and Wikipedia distance with dimension reduction.

Table 4 Dataset characteristics.

Pages in dataset	100
Ads in dataset	7,996
Page-ad judgments	4,406
Wikipedia articles	1,000

5 Experiments and results

5.1 Data and methodology

We conducted experiments to evaluate our method using a dataset containing 100 pages and 7,996 ads as shown in Table 4. For each page, we collected human judgment scores that evaluate the relevance of selected ads by each of the compared methods. Selected ads on each page are marked by human judges as relevant or not relevant. Because the purpose of ads matching is to select the top-N of relevant ads for a given page, we evaluated the average precision for the top-1, top-3 and top-5 (usually the number of ads displayed on a Web page is not greater than 5):

$$\text{AveP}(K) = \frac{\sum_{\forall p \in P} N(\text{relevant ads})}{K \cdot N(\text{pages})}$$

where P is the set of pages, K is a number of retrieved ads (i.e. $K = 1$ for top-1, $K = 3$ for top-3, $K = 5$ for top-5), N(relevant ads) is a number of relevant ads according to the human judgment scores. We also consider the sum of those precisions as an overall score:

$$\text{SumAveP} = \text{AveP}(1) + \text{AveP}(3) + \text{AveP}(5)$$

The same evaluation method was used in [14] and [3].

5.2 Determining aggregation method

First, we will determine the best aggregation method for our proposed techniques: Wikipedia similarity and Wikipedia distance. We run the experiments and compare precisions at top-1, top-3, top-5 and the overall score. From the results in Table 5, we can see that Borda's count gives a better performance for the Wikipedia similarity than the weighted sum. The corresponding rows (the ones with higher values) are marked with a bold font. Results in Table 6 show that for the Wikipedia distance, the weighted sum works better. Therefore in our next experiments, we will consider methods: BKW, KE1 and KE2.

Table 5 Comparing the weighted sum and the Borda's count for the Wikipedia similarity.

Method	Top-1	Top-3	Top-5	Sum
KW	0.705	0.74	0.705	2.151
BKW	**0.737**	0.74	**0.726**	**2.204**

Table 6 Comparing the weighted sum and the Borda's count for the Wikipedia distance in the original and reduced dimension space.

Method	Top-1	Top-3	Top-5	Sum
KE1	0.8	**0.754**	**0.705**	**2.26**
BKE1	0.8	0.709	0.665	2.174
KE2	**0.821**	**0.744**	**0.714**	**2.279**
BKE2	0.737	0.691	0.669	2.098

5.3 Average precision

Next, we compare the performance of different matchings methods. We will consider the following methods:

- Traditional keyword matching (K)
- Semantic-syntactic matching (KS)
- Wikipedia similarity with keyword matching (BKW)
- Wikipedia distance with keyword matching (KE1)
- Wikipedia distance after dimension reduction with keyword matching (KE2)
- The combination of all the considered methods (by the weighted sum and the Borda's count) (ALL, BALL)

The result of the average precision for all seven strategies is depicted in Figure 5. As we can observe from the graphs, the best matching is achieved by *ALL* combination that aggregates semantic-syntactic matching and our proposed Wikipedia matching methods. The next best result is obtained by KE2, that is using Wikipedia distance with dimension reduction. Then goes the KE1—Wikipedia distance, BALL—the combination of all considered methods aggregated by Borda's count, KW—Wikipedia similarity, KS—semantic-syntactic approach, and finally the traditional keyword matching. We state that our proposed Wikipedia matching allows to improve the performance of existing solutions significantly. The Borda's aggregation of all the methods can be used when it is not possible to tune parameters regularly.

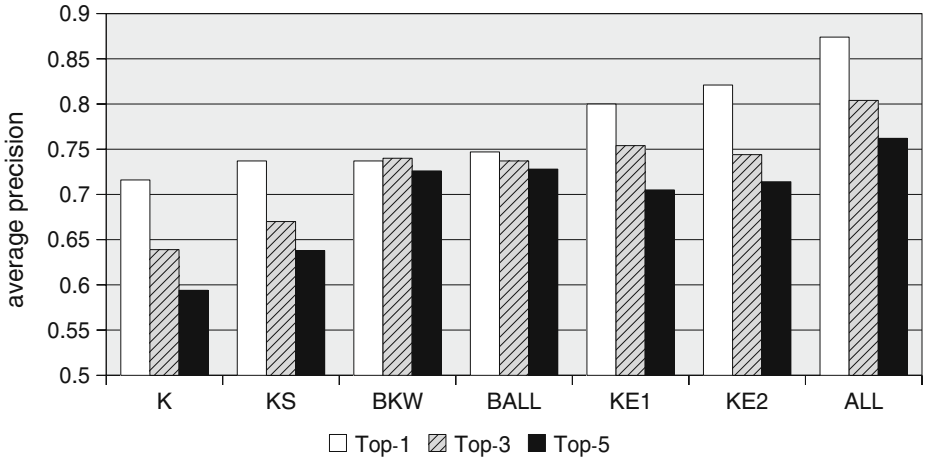

Figure 5 Average precision over all pages.

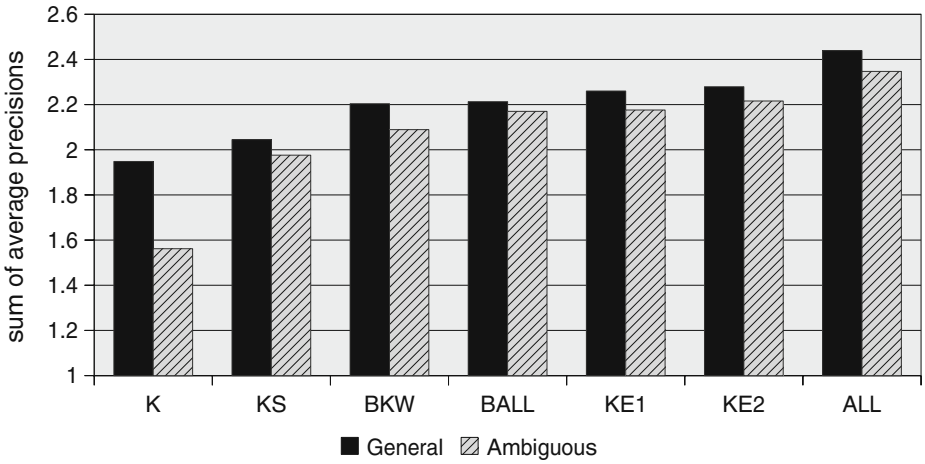

Figure 6 Sum of average precisions for the common dataset and the ambiguous dataset.

5.4 Results for the ambiguous dataset

To show that our proposed method helps to overcome the problems of keyword matching, we have selected a special dataset consisting of ambiguous pages. These are pages that contain ambiguous keywords or ambiguous context. We show the list of examples of such pages in Table 7. The results for experiments run on the ambiguous dataset is presented in Figure 6. As we can see from the graph, our proposed solutions perform well on both of datasets, while the performance of keyword matching degrades a lot. Thus, we conclude that by using our proposed techniques: Wikipedia similarity and Wikipedia distance, we can reduce the negative effect caused by the problems of traditional keyword matching.

5.5 Performance gain and t-interval

We performed a statistical analysis of paired samples using the t-statistics [2] to prove that there is an evidence that our method is better than traditional keyword matching and to obtain 99.9% two-sided confidence interval for the performance gain. We compare two populations: sums $(x_1, x_2, \ldots, x_n)$ of precision values obtained by keyword matching for each pages and corresponding sum values $(y_1, y_2, \ldots, y_n)$ by using compared methods for each pages. We take the following steps to find the t-interval:

- First, we obtain a performance gain value for each page: $z_i = y_i - x_i \ \ \forall i \in [1, n]$, where n is the number of pages
- Next, we assume that mean value $\bar{z}$ should be within a interval: $(\bar{z} - \epsilon, \bar{z} + \epsilon)$, where ϵ is an error, a random variable following a normal distribution
- According to the definition of two-sided t-interval, the value of ϵ is calculated as: $\epsilon = \frac{t_{\alpha/2,n-1} s}{\sqrt{n}}$ where $t_{\alpha/2,n-1}$ is a critical point of Student's t-distribution, α is fixed value (we use $\alpha = 0.001$ which corresponds to 99.9% of confidence), and s is the standard deviation of z

Table 7 An example list of ambiguous pages.

Page title	Source of ambiguity
Nigeria: panic as HIV/Aids spreads among workers	"AIDS" vs. "aid"
Gas prices fall near the $1.80 mark	"gas" (oil) vs. natural "gas"
Hospital doctor in Burress incident suspended	"hospital" vs. "hospitality"
Business | rock delays moves on repossession	"Rock" (a person) vs. "rock" (music)
US seeks urgent action on Mumbai	"Rice" (a person) vs. "rice" (grain)
Windows 7 beta in January?	"Windows" vs. "window"
BlackBerry sign-ups short of goal	(weather) "forecast" vs. (financial) "forecast"
Jupiter and Venus have been teaming up in sky lately	"Jupiter", "Venus" (brands vs. planets)
Vitamin D vital for the heart	"heart" (organ) vs. "heart" (center)
Reports say Charlie Weis staying at Notre Dame	"game" (sports) vs. (computer) "game"
White Sox send Vazquez to Braves	"Flowers" (a person) vs. "flower"
Treatments help turn AIDS into manageable disease	"cell" (biology) vs. "cell" (electronics)
Hitachi/Intel push solid state drives forward	"drive" (computers) vs. "drive" (cars)
UCLA's Kevin craft takes a beating without bleating	"bowl" (championship) vs. "bowling"
YouTube's got an ear (and eye) for music	"phone" (mobile vs. stationary)
Mandatory testing and treatment can end the AIDS epidemic	"AIDS" vs. "aid"
Car cell phone use more hazardous than chat with passengers	(car) "driver" vs (software) "driver"

The results of obtained t-intervals are presented on Figure 7 for the common dataset and on Figure 8 for the ambiguous dataset. As we can see from the graph, the most gain in precision is obtained by the ALL combination. For the common dataset it allows to achieve up to 24% of performance gain, and more than 50% on the ambiguous dataset.

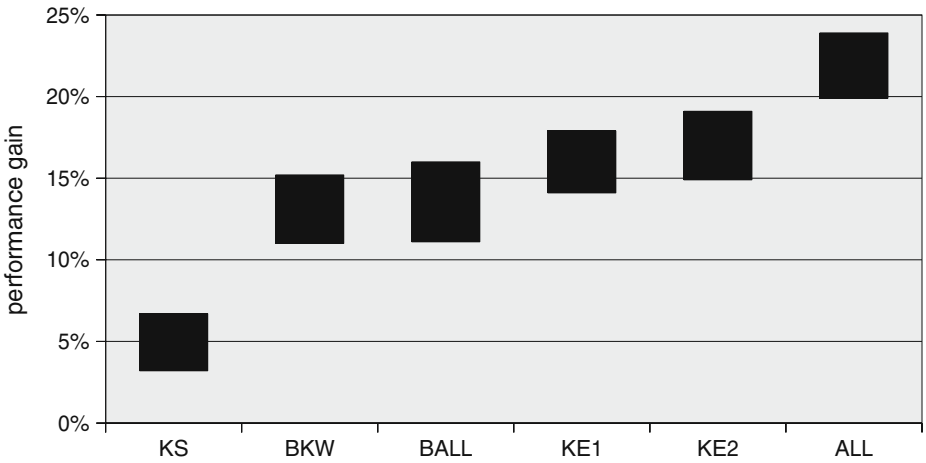

Figure 7 99.9% two-sided confidence interval for performance gain.

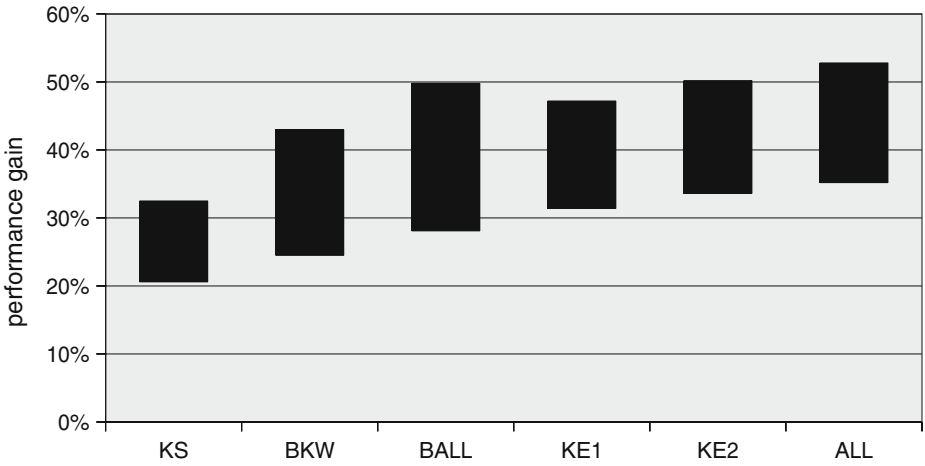

Figure 8 99.9% two-sided confidence interval for performance gain on the ambiguous dataset.

5.6 Binary preference measure

To confirm our results, we use an additional performance measure — binary preference (bpref) [4, 17]. Bpref is a TREC[10] standard measure for partially evaluated document collections. Bpref measures whether judged relevant documents have higher scores than judged irrelevant documents:

$$\text{bpref} = \frac{1}{R}\sum_{r} 1 - \frac{|n \text{ ranked higher than } r|}{R} \tag{13}$$

where R is the number of judged relevant documents, r is a relevant retrieved document, and n is a member of the first R irrelevant retrieved documents.

We use a variant of bpref — bpref-10 [4], which is considered to be more stable than bpref.

$$\text{bpref-10} = \frac{1}{R}\sum_{r} 1 - \frac{|n \text{ ranked higher than } r|}{R+10} \tag{14}$$

We use top-10 ads on each page to obtain bpref-10 scores and then averaged them over all pages.

Bpref-10 evaluation is presented in Figure 9. The obtained results confirm that using Wikipedia matching improves the precision of ads selection.

5.7 Impact of the number of articles

We have tested the impact of the number of articles used in Wikipedia matching on the relevance of ads selection. We assumed that the number of articles should be large enough to make a good distinction of ads for a given page. Thus, we performed

[10]Text REtrieval Conference— http://trec.nist.gov/.

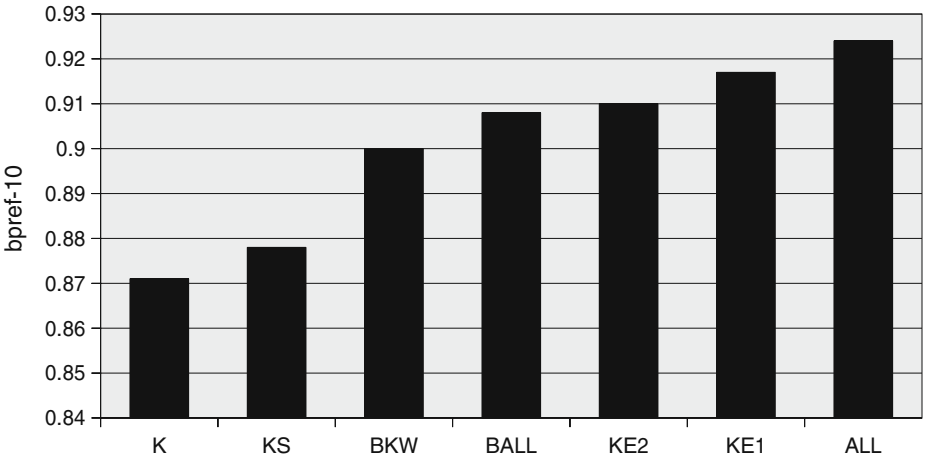

Figure 9 Bpref-10 metrics.

an experiment, in which we changed the number of used articles (from 100 to 2,000) and observed the sum of average precisions of selected ads at top-1, top-3 and top-5. The result for the Wikipedia similarity method is presented in Figure 10. As we expected, the precision of ads selection improves when increasing the number of articles. However at some moment, there is enough articles to make a relevant selection of ads and a further improvement of ads selection by adding more articles is not feasible. We can observe in the graph that the precision improves notably when we increase the number of articles from 100 to 1,000, but when increasing the number from 1,000 to 2000, the improvement is very small. Further increasing of the number of articles is not reasonable, because it slows down the system's performance.

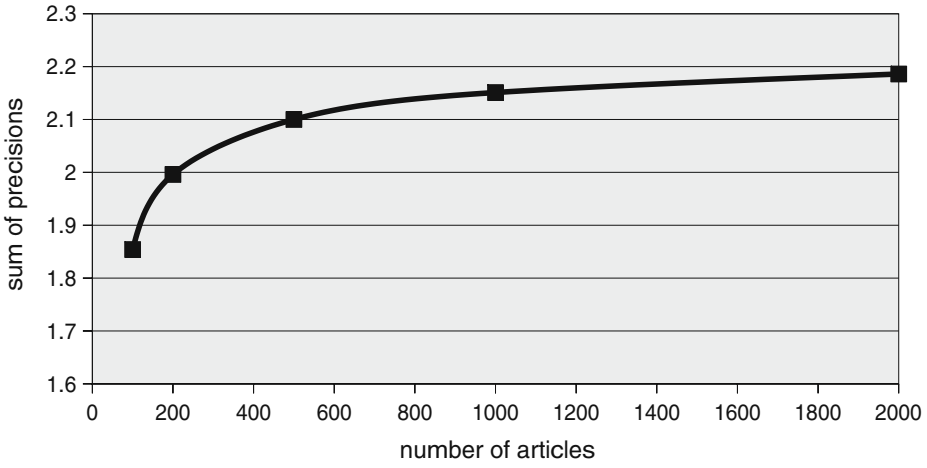

Figure 10 The impact of the number of articles on the sum of average precisions.

6 Conclusion

We have proposed a new strategy for matching contextual ads. Prior works based on keyword matching have problems caused by homonymy and polysemy, the low intersection of keywords, and the context mismatch. The previously proposed approach of semantic-syntactic matching is sensitive to the document classification precision and thus needs to be improved. Our matching technique uses Wikipedia articles as reference points to establish matching between ads and pages. We have proposed two ranking methods for selecting ads. First is called Wikipedia similarity and it considers Wikipedia articles as classification labels. The second is called Wikipedia distance, which uses the Euclidean distance in a multi-dimensional space constructed by considering articles as semantic dimensions. The latter technique also can be improved by applying a dimension reduction technique—the principal component analysis.

Our method can be used in combination with other approaches. The best strategy combines semantic-syntactic matching proposed by Broder et. al with our proposed ranking functions: Wikipedia similarity, Wikipedia distance and Wikipedia distance in the reduced dimensions. Experimental evaluations show that our proposed Wikipedia matching improves the precision of selected ads by traditional keyword matching and semantic-syntactic matching strategies. A statistical t-test was used to confirm that our proposed method performs better than previous solutions. We have also confirmed the positive effect of using Wikipedia matching by applying TREC standard measure bpref-10.

As a future work we plan to use Wikipedia matching with other existing solutions and evaluate the obtained effect. We also plan to exploit machine learning techniques for estimating parameter values of our algorithm.

Acknowledgements This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (grant number 2009-0081365).

References

1. Anagnostopoulos, A., Broder, A., Gabrilovich, E., Josifovski, V., Riedel, L.: Just-in-time contextual advertising. In: Proc. of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 331–340, Lisbon, Portugal (2007) L
2. Anthony H.J.: Probability and Statistics for Engineers and Scientists. Duxbury, Belmont (2007)
3. Broder, A., Fontoura, M., Josifovski, V., Riedel, L.: Semantic approach to contextual advertising. In: Proc. of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands (2007)
4. Buckley, C., Voorhees, E.M.: Retrieval evaluation with incomplete information. In: SIGIR '04: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 25–32, New York, NY, USA, ACM (2004)
5. Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 491–407 (1988)
6. Ding, C., He, X.: K-means clustering via principal component analysis. In: ICML '04: Proceedings of the Twenty-first International Conference on Machine Learning, p. 29, New York, NY, USA, ACM (2004)
7. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: WWW '01: Proceedings of the 10th International Conference on World Wide Web, pp. 613–622, New York, NY, USA, ACM (2001)

8. IDC.: Worldwide and U.S. Internet ad Spend Report 4q08: U.S. Growth Flat, 1q09 Spending Likely to Contract (2009)
9. Jun, Z.: Comprehensive Perl Archive Network. http://search.cpan.org/~jzhang/html-contentextractor-0.02/lib/html/contentex%tractor.pm (2007)
10. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
11. Murdock, V., Ciaramita, M., Plachouras, V.: A noisy-channel approach to contextual advertising. In: Proc. of the 1st International Workshop on Data Mining and Audience Intelligence for Advertising, pp. 21–27, San Jose, California (2007)
12. Porter, M.F.: An algorithm for suffix stripping. Readings in Information Retrieval, pp. 313–316 (1997)
13. Porter, M.F.: The Porter Stemming Algorithm official home page. http://tartarus.org/~martin/porterstemmer/index.html (2006)
14. Ribeiro-Neto, B., Cristo, M.: Impedance coupling in content-targeted advertising. In: Proc. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 496–503, Salvador, Brazil (2005)
15. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM, **18**(11), 613–620 (1975)
16. Sullivan, D.: Search Engine Watch. http://searchenginewatch.com/2183531 (2003)
17. TREC.: The Fifteenth Text Retrieval Conference (TREC 2006) Proceedings. http://trec.nist.gov/pubs/trec15/appendices/ce.measures06.pdf (2006)
18. Zhang, Y., Vogel, S.: Measuring confidence intervals for the machine translation evaluation metrics. In: In Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation, TMI-2004, pp. 4–6 (2004)