Just-in-Time Contextual Advertising

Aris Anagnostopoulos, Andrei Z. Broder, Evgeniy Gabrilovich, Vanja Josifovski, Lance Riedel
Yahoo! Research, 2821 Mission College Blvd, Santa Clara, CA 95054
{aris | broder | gabr | vanjaj | riedell}@yahoo-inc.com

ABSTRACT

Contextual Advertising is a type of Web advertising, which, given the URL of a Web page, aims to embed into the page (typically via JavaScript) the most relevant textual ads available. For static pages that are displayed repeatedly, the matching of ads can be based on prior analysis of their entire content; however, ads need to be matched also to new or dynamically created pages that cannot be processed ahead of time. Analyzing the entire body of such pages on-the-fly entails prohibitive communication and latency costs. To solve the three-horned dilemma of either low-relevance or high-latency or high-load, we propose to use text summarization techniques paired with external knowledge (exogenous to the page) to craft short page summaries in real time. Empirical evaluation proves that matching ads on the basis of such summaries does not sacrifice relevance, and is competitive with matching based on the entire page content. Specifically, we found that analyzing a carefully selected 5% fraction of the page text sacrifices only 1%–3% in ad relevance. Furthermore, our summaries are fully compatible with the standard JavaScript mechanisms used for ad placement: they can be produced at ad-display time by simple additions to the usual script, and they only add 500–600 bytes to the usual request.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation, Selection process*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)*

General Terms: Algorithms, Experimentation, Measurement, Performance

Keywords: Text classification, text summarization

1. INTRODUCTION

The total Internet advertiser spending in 2006 in the US alone is estimated at over 17 billion dollars, with a growth rate of almost 20% year over year. A large part of this market consists of *textual ads*, that is, short text messages usually marked as "sponsored links" or similar. Today, there are two main types of textual Web advertising: *sponsored search*, which serves ads in response to search queries, and *content match*, which places ads on third-party Web pages. In the former case, ads are matched to the (short) query issued by the user, and in the latter case ads are matched to the entire page content. In both cases, it has been shown that the response of the users to the advertising is related to how *relevant* the ads are to the query or to the page (respectively).

In this paper, we study a very common content match scenario, where Web site owners (called *publishers*) provide the "real-estate space" (i.e., a reserved portion of their page) for placing ads, and the ad server or *ad network*, an entirely different commercial entity, returns the ads that are most suitable for the page content. Typically, this is done via JavaScript: the display of the page on a user's screen results in calls being made to the ad server for the supply of suitable textual ads. These calls provide the URL of the page being displayed, and potentially other data.

When a user requests to view a page, the ad selection engine has only a couple hundred milliseconds to provide the ads. In most cases this low latency requirement does not allow for pages to be fetched and analyzed online. Instead, the pages are fetched and analyzed offline, and the results are applied in subsequent ad serving for the same page. This approach works well for static content pages that are displayed repeatedly.

However, a significant amount of the Web is not static: some pages are dynamic by definition, such as personalized pages, and the front pages of news sites, forums, and blogs are constantly changing. Some pages cannot be accessed in advance because they belong to the "invisible Web," that is, they do not exist, except as a result of a user query. Yet other pages are not independently accessible since they require authorizations and/or cookies that are present on the user's computer but not on the ad server's platform. In all of these examples, ads need to be matched to the page *while it is being served to the end-user*, thus critically limiting the amount of time allotted for its content analysis.

Thus, our challenge is to find relevant ads while maintaining low latency and communication costs. We propose a two-pronged approach to solve it:

1. We employ text summarization techniques to extract short but informative excerpts of page text that are representative of the entire page content. In addition to these excerpts, we also use the information in the

CIKM'07, November 6–8, 2007, Lisboa, Portugal.
Copyright 2007 ACM 978-1-59593-803-9/07/0011 ...$5.00.

page URL, as well as the referrer URL. All this data can be produced by the JavaScript code as the page is being displayed, and only the summary information is sent to the ad server.

2. In line with our previous work on full pages [5], we classify the page summaries with respect to a large taxonomy of advertising categories, and perform page-ad matching based on both bag of words features and classification features.

The volume of pages in contextual advertising systems follows the long tail (power law) model, where a relatively small number of pages are seen numerous times and the majority of pages are seen only a few times. In addition to eliminating the need for re-crawls of static pages, our approach also reduces the need for crawling "tail" pages that are rarely seen by the system. If the page content can be analyzed using a serving-time summary, it might not be necessary (nor economically viable) to crawl the page ahead of time. This would limit the crawling only to the pages in the head and the torso of the volume curve, and therefore save additional networking and processing resources both for the ad server and for the publisher.

Previous studies have explored content match based on different ad parts (see Section 5 for a full discussion). While selecting the right ad parts to perform the match is certainly important from the relevance point of view, ads are available beforehand, and so their leisurely analysis has no impact on latency. Here we focus on analyzing the information content of the different page parts, at *ad-display time*, when communication and processing time are at a premium.

The main contributions of this paper are threefold. First, we describe a novel method that enables online contextual matching of pages and ads. We create a concise page summary on the fly, and match ads based on this summary rather than the entire page. Empirical evaluation confirms that matching ads based on dynamically created page summaries yields ads whose relevance is on par with that of the full page analysis. Second, we analyze the role and the feasibility of semantic match of the page and the ads based on text classification of page excerpts and ads. Third, our findings imply that frequent repeated crawling of publisher pages can be avoided by analyzing page summaries just in time for actual page display. Consequently, our method reduces system load by making it unnecessary to crawl numerous "tail pages," and allows to serve relevant ads for dynamically changing pages.

The rest of the paper is organized as follows. Section 2 provides background on current practices in Web advertising. Section 3 outlines our methodology for robust page analysis. Empirical evaluation of our methodology is presented in Section 4. We survey the related work in Section 5. We discuss our findings and draw conclusions in Section 6.

2. WEB ADVERTISING BASICS

In this section we give a brief overview of the current practices in Web advertising, which is based on a longer presentation in our earlier work [5].

A large part of the Web advertising market consists of *textual ads*, which are distributed through two main channels:

1. *Sponsored Search* or *Paid Search Advertising*, which places ads on the result pages of a Web search engine, with ads being driven by the original query. All major Web search engines (Google, Microsoft, Yahoo!) support sponsored ads and act simultaneously as a web-search engine and an ad-search engine.
2. *Content Match* (CM) or *Contextual Advertising*, which places commercial ads on any given Web page (see [10] for a brief history of the subject). Today, almost all of the for-profit non-transactional web sites[1] rely at least to some extent on advertising revenue. Content match supports sites that range from individual bloggers and small niche communities to large publishers such as major newspapers. Without this model, the web would be a lot smaller!

Contextual advertising is an interplay of the following four entities:

- The **publisher** is the owner of Web pages on which advertising is displayed. The publisher typically aims to maximize advertising revenue while providing a good user experience.
- The **advertiser** provides the supply of ads. Usually the activity of the advertisers is organized around *campaigns*, which are defined by a set of ads with a particular temporal and thematic goal (e.g., sale of digital cameras during the holiday season). As in traditional advertising, the goal of the advertisers can be broadly defined as promotion of products or services.
- The **ad network** is a mediator between the advertiser and the publisher, who selects the ads that are put on the pages. The ad-network shares the advertising revenue with the publisher.
- **Users** visit the Web pages of the publisher and interact with the ads.

Given a page, instead of placing generic ads, it is preferable to have ads related to the page content in order to provide a better user experience and to increase the probability of clicks. This intuition is supported by the analogy to conventional publishing, where a number of very successful magazines (e.g., *Vogue*) have a majority of the pages devoted to topical advertising (fashion in the case of *Vogue)*. A number of user studies also confirmed that improved relevance increases the number of ad-clicks [8, 27].

Contextual advertising usually falls into the category of *direct marketing* (as opposed to *brand advertising*), that is, advertising whose aim is a "direct response," where the effect of a campaign is measured by the user reaction (e.g., purchase of advertised goods or services). Compared to the traditional media, one of the advantages of online advertising in general and contextual advertising in particular is that it is relatively easy to measure the user response. Usually the desired immediate reaction is for the user to follow the link in the ad and visit the advertiser's Web site.

The prevalent pricing model for textual ads is that the advertisers pay a certain amount for every click on the advertisement (pay-per-click or PPC). There are also other models, such as pay-per-impression, where the advertiser pays for the number of exposures of an ad, and pay-per-action, where the advertiser pays only if the ad leads to a sale or similar completed transaction. In this paper we deal with the PPC model.

[1] Non-transactional sites are those that do not sell anything directly.

Content match advertising has grown organically from sponsored search advertising. In most networks, the amount paid by the advertiser for each sponsored search click is determined by an auction process. The advertisers place bids on a search phrase, and their position in the tower of ads displayed on the search results page is determined by their bid. Thus, each ad is annotated with one or more *bid phrases*. The bid phrase has no direct bearing on the ad placement in content match. However, it is a concise description of target ad audience as determined by the advertiser, and it has been shown to be an important feature for successful CM ad placement [19]. In addition to the bid phrase, an ad is also characterized by a *title* usually displayed in bold font, and an *abstract* or *creative*, which is the few lines of text, usually shorter than 120 characters, displayed on the page. Naturally, each ad contains a URL to the advertised Web page, called *landing page*.

The ad-network model aligns the interests of the publishers, advertisers and the network. In general, clicks bring benefits to the publisher and the ad network by providing revenue, and to the advertiser by bringing traffic to the target web site. The revenue of the network, given a page p, can be estimated as

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

where k is the number of ads displayed on page p and $price(a_i, i)$ is the click-price of the given ad a_i at position i. The price in this model depends on the set of ads presented on the page. Several models have been proposed to determine the price, most of them based on generalizations and variants of second price auctions. In this paper, we ignore the pricing model for simplicity, and concentrate on finding ads that will maximize the first term of the product, that is, we search for

$$\arg\max_i P(click|p, a_i).$$

Furthermore, we assume that the probability of a click for a given ad and page is determined by the ad's relevance score with respect to the page, thus ignoring the positional effect of the ad placement on the page. We assume that this is an orthogonal factor to the relevance component and could be easily incorporated in the model.

3. METHODOLOGY

In this section we first define in more detail the problem of efficiently matching ads to pages, and then develop the proposed solution.

3.1 Problem Statement

The typical content match approach for displaying ads on Web pages is outlined in Figure 1. Upon a request initiated by the user's browser (HTTP **get** request), the Web server returns the requested page. As the page is being displayed, a JavaScript code embedded into the page (or loaded from a server) sends to the ad server a request for ads that contains the page URL and possibly some additional data.

When the page contents is static (that is, the content associated to the given URL is not generated on-the-fly and changes infrequently), the ad server can invest computation resources in a one-time offline process that involves fetching the entire page and performing deep analysis of the page content to facilitate future ad matches. However, ads need to be matched also to new or dynamically created pages that cannot be processed ahead of time, and analyzing the entire body of such pages at display-time entails prohibitive communication and latency costs.

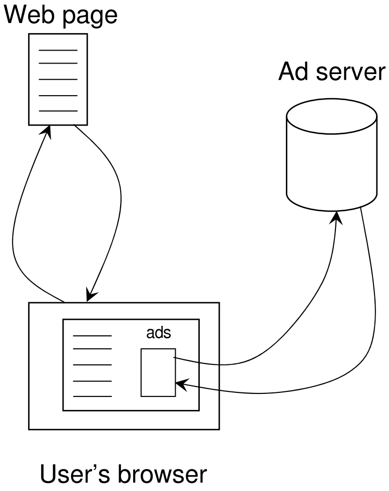

Figure 1: Overview of ad display

If the page content cannot be analyzed in advance, we are facing a three-horned dilemma:

- **Low-relevance ads.** We can serve generic ads that are unrelated to the page actual content (sometimes these ads are called *run-of-network* or *RON* ads). However, these ads are seldom appealing to users, thus resulting in fewer clicks; furthermore, these ads are sold at lower PPC than matched ads.
- **High communication and preprocessing load.** We can crawl every ad-displaying page very frequently, so that the ad server has a recent snapshot of its content. In the extreme case, the ad server can retrieve the page every time there is an ad request for that page. This would, of course, double the load on publisher's server. This option not only creates an excessive load on both the publisher's server and the ad server, but in many cases it is not feasible at all—some pages are only generated upon a parameterized request, and it is impossible to pre-crawl all the pages corresponding to all possible combination of parameters. This option is also not available for pages that require authorizations and/or cookies that are present on the user's computer but not on the ad server's platform.
- **High latency.** The JavaScript used to request ads can be used to send the entire content of the page being displayed to the ad server. In turn, the ad server can then analyze the entire content of the page and return the most relevant ads available. This approach significantly increases the amount of communication between the user's browser and the ad server, as well as the processing load on the ad server, resulting in a long delay until the ads can be displayed. This leads to poor user experience, and in fact the user might be gone before the ads have even arrived.

Thus, our challenge is to produce highly relevant ads without any pre-crawling of Web pages, using only a modest amount of processing and communication resources at ad-display time.

3.2 Overview of the Proposed Solution

Our solution is to use text summarization techniques paired with external knowledge (exogenous to the page) to craft short page summaries in real-time. The summaries are produced within the standard JavaScript mechanisms used for ad placement and they only add 500–600 bytes to the usual request. Thus, our approach balances the two conflicting requirements: analyzing as much page content as possible for better ad match vs. analyzing as little as possible to save transmission and analysis time.

For summaries, we use several techniques [7, 15] to extract short but concise page excerpts that are highly informative of the entire page content.

To supplement the page summary, we also use external knowledge from a variety of sources, namely:

1. **URL.** We tokenize the page URL into individual words, on the premise that page URLs often contain meaningful words that are relevant to the page content.
2. **Referrer URL.** We also analyze the referrer URL, that is, the URL from where the user arrived to the current page (the referrer URL is available in the JavaScript). This URL might contain relevant words that to some extent capture the user intent, for instance, if the referrer was a hub or a search result page.
3. **Page classification.** More importantly, we classify the page content onto a large taxonomy and use resultant classifications to augment page representation for ad matching. To this end, we pre-classify all the ads onto the same taxonomy, and then perform the matching in the extended space of word-based and classification-based features as opposed to the plain bag of words.

   Intuitively, one would opt to classify the entire page, but doing so would incur high transmission and processing costs as explained above. However, it was previously found [15, 23] that text summarization can be successfully used as a preprocessing step for classification. Indeed, we choose to classify the page summary instead of the full page. As we show in Section 4, our results corroborate previous findings, and in many cases the results of classifying a succinct summary are competitive with full-page classification. We also show that using taxonomy-based classification has measurable positive effect on ad relevance.

One often used source of external knowledge about Web pages is anchor text of incoming links [3]. However, we do not use such anchor text in this work since in many cases advertisement pages are dynamic, and therefore have no anchor text. Furthermore, our just-in-time approach can also be used to put relevant ads on new pages, for which little or no anchor text is available.

In the experiments reported in Section 4, our baseline corresponds to matching ads by analyzing the full text of the page (including the page and referrer URLs, as well as the classification information). We use a variety of text summarization techniques to achieve substantial reduction in processing time while demonstrating matching relevance that is on par with (or even better than) full page analysis.

3.3 The Nuts and Bolts

We now explain our methodology in more detail.

3.3.1 Text Summarization

Text summarization techniques are divided into *extractive* and *non-extractive* approaches. The former approach strives to summarize the document by taking carefully selected terms and phrases that are already present in the document. The latter approach analyzes the entire document as a whole and rewrites its content in a more concise way; this option is usually very resource- and computation-intensive, hence we adopt the extractive approach.

Since our input is an HTML document, we rely on the HTML markup that provides hints to the relative importance of the various page segments. This allows us to avoid time-consuming analysis of the text by taking cues from the document structure. When the user's browser displays the Web page, it actually performs HTML parsing prior to rendering, hence the JavaScript code embedded into the page has easy access to the DOM[2] representation of the parsed document.

Following prior works [7, 15], we evaluate the role of the following page components in constructing summaries:

- Title (**T**)
- Meta keywords and description (**M**)
- Headings (**H**): the contents of `<h1>` and `<h2>` HTML tags, as well as captions of tables and figures
- Tokenized URL of the page (**U**)
- Tokenized referrer URL (**R**)
- First N bytes of the page text (e.g., $N = 500$) (**P<N>**, e.g., **P500**)
- Anchor text of all outgoing links on the page (**A**)
- Full text of the page (**F**)

In the next section, we evaluate the individual contribution of each of the above-listed page segments as well as their combinations for serving a page proxy for ad matching. To tokenize URLs into words, we used a dynamic programming tool developed in-house, which relied on a language model built from a corpus of several million documents.

3.3.2 Text Classification

Using a summary of the page in place of its entire content can ostensibly eliminate some information. To alleviate possible harmful effect of summarization, we study the effects of using external knowledge by means of classifying page summaries with respect to an elaborate taxonomy. Prior studies found that text summarization can actually improve the accuracy of text classification [15, 23]. A recent study also found that features constructed with the aid of a knowledge-based taxonomy are beneficial for text classification [11]. Consequently, we classify both page excerpts and ads with respect to a taxonomy, and use classification-based features to augment the original bag of words in each case.

Choice of Taxonomy.

Our choice of taxonomy was guided by a Web advertising application. Since we want the classes to be useful for matching ads, the taxonomy needs to be elaborate enough to facilitate ample classification specificity. For example, classifying all medical queries into one node will likely result

[2] Document Object Model (DOM) is a standard approach to representing HTML/XML documents [26].

in poor ad matching, as both "sore foot" and "flu" queries will end up in the same node. The ads appropriate for these two queries are, however, very different. To avoid such situations, the taxonomy needs to provide sufficient discrimination between common commercial topics. Therefore, we employed a large taxonomy of approximately $6,000$ nodes, arranged in a hierarchy with median depth 5 and maximum depth 9. Human editors populated the taxonomy with labeled bid phrases of actual ads (approx. 150 phrases per node), which were used as a training set; a small fraction of queries have been assigned to more than one category. We used the same taxonomy in our earlier work [4], where it is described in more detail.

Classification Method.

Few machine learning algorithms can efficiently handle so many different classes and about an order of magnitude more of training examples. Suitable candidates include the nearest neighbor and the Naive Bayes classifier [9], as well as prototype formation methods such as Rocchio [21] or centroid-based [12] classifiers.

We used the latter method to implement our text classifier. For each taxonomy node we concatenated all the phrases associated with this node into a single meta-document. We then computed a centroid for each node by summing up the *TFIDF* values of individual terms, and normalizing by the number of phrases in the class:

$$\vec{c_j} = \frac{1}{|C_j|} \sum_{\vec{p} \in C_j} \frac{\vec{p}}{\|\vec{p}\|},$$

where $\vec{c_j}$ is the centroid for class C_j and p iterates over the phrases in a particular class.

The classification is based on the cosine of the angle between the document and the centroid meta-documents:

$$C_{max} = \arg\max_{C_j \in C} \frac{\vec{c_j}}{\|\vec{c_j}\|} \cdot \frac{\vec{d_j}}{\|\vec{d_j}\|}$$

$$= \arg\max_{C_j \in C} \frac{\sum_{i \in |F|} c^i \cdot d^i}{\sqrt{\sum_{i \in |F|} (c^i)^2} \sqrt{\sum_{i \in |F|} (d^i)^2}},$$

where F is the set of features, and c^i and d^i represent the weight of the ith feature in the class centroid and the document, respectively. The scores are normalized by the document and centroid lengths to make the scores of different documents comparable. These weights are based on the standard "ltc" *TFIDF* function [22].

Using Classification Features.

We classified each page summary and each ad with respect to the taxonomy, retaining the 5 top-scoring classifications for each text fragment. Following [11], we constructed additional features based on these immediate classifications as well as their ancestors in the taxonomy (the weight of each ancestor feature was decreased with a damping factor of 0.5). Each page and ad were represented as a bag of words (BOW) and an additional vector of classification features. Finally, the ad retrieval function was formulated as a linear combination of similarity scores based on both BOW and classification features:

$$score(page, ad) = \alpha \cdot sim_{BOW}(p, a) + \beta \cdot sim_{class}(p, a),$$

where $sim_{BOW}(p, a)$ and $sim_{class}(p, a)$ are cosine similarity scores between page p and ad a using BOW and classification features, respectively.

4. EMPIRICAL EVALUATION

We start with the description of the dataset and the metrics used, and then proceed to discuss the experimental results. Unless specified otherwise, all the experiments below employ both text summarization and text classification techniques; the effect of text classification in isolation is studied in Section 4.7.

4.1 Datasets

To evaluate the effects of text summarization and classification for efficient ad matching, we used two sets of Web pages, which have been randomly selected from a larger set of around 20 million pages with contextual advertising. Ads for each of these pages have been selected from a large pool of about 30 million ads. We preprocessed both pages and ads by removing stopwords and one-character words, followed by stemming. We collected human judgements for over 12,000 individual page-ad pairs, while each pair has been judged by three or more human judges on a 1 to 3 scale:

1. **Relevant** The ad is semantically directly related to the main subject of the page. For example, if the page is about the National Football League and the ad is about tickets for NFL games, this page-ad pair would be scored as 1.
2. **Somewhat relevant** The ad is related to the secondary subject of the page, or is related to the main topic of the page in a general way. For example, given an NFL page, an ad about NFL-branded products would be judged as 2.
3. **Irrelevant** The ad is unrelated to the page. For example, a mention of the NFL player John Maytag triggers ads for Maytag-manufactured washing machines on a NFL page.

To obtain a single score for a page-ad pair, we averaged the human judgments. We then used these judgments to evaluate how well our methods distinguish the positive (relevant) and the negative (irrelevant) ad assignments for each page. An ad is considered relevant if its score is below some threshold, otherwise it is irrelevant. We experimented with several different thresholds (ranging between 1.7–2.4), and found that they did not affect the conclusions. In all the graphs presented below we used the threshold of 2.4 (i.e., most of the judges considered the ad somewhat relevant). Based on human judgments, we eliminated pages for which the judged ads were all relevant or all irrelevant (after the thresholding procedure), as they provide little information in judging different algorithmic ad rankings.

The two sets of pages we used are inherently different. Dataset 1 consists of Web pages that are accessible through a major search engine, and have actually appeared in the first 10 results for some query; consequently, they tend to be of better quality with more textual content. On the other hand, Dataset 2 consists of pages from publishers that are not found in the search engine index, and therefore are generally of lower quality with less text and more images and advertising. Having these two datasets allows us to evaluate our methodology in a more comprehensive way. The statistics for the two datasets are given in Table 1. The pages

Page fragment		Dataset 1		Dataset 2	
Description	Shorthand	Avg. size (bytes)	Num. pages	Avg. size (bytes)	Num. pages
Page HTML	–	36,525	200	32,118	1,756
Full text	F	8,508	198	6,779	1,739
Anchor text	A	1,525	192	1,140	1,556
First 500 bytes	P500	495	198	431	1,699
Title	T	55	198	45	1,751
Meta data	M	267	162	411	1,074
Headings	H	109	116	65	505
Page URL	U	48	200	–	–
Referrer URL	R	40	137	–	–

Table 1: Sizes of page fragments

in Dataset 1 have more textual content than in Dataset 2. In addition to the amount of text, visual inspection of the pages indicates that the content on the pages in Dataset 1 is much more consistent around the page topic. Furthermore, pages in Dataset 1 have on average twice as many judgments as in Dataset 2 (28 vs. 11.7). For these reasons, and due to space paucity, we emphasize Dataset 1 in our evaluation.

4.1.1 Dataset 1

Dataset 1 consisted of 200 Web pages of various types, ranging from `Amazon.com` query result pages to medical documents, Wikipedia articles, online tutorials, and so on. Upon eliminating pages for which all judged ads had identical scores (as explained above), we ended up with a set of 105 pages that were used in the experiments. There were 2,680 unique ads and 2,946 page-ad scores (some ads have been scored for more than one page). Inter-judge agreement in scoring was 84%. We classified the pages and ads as explained in Section 3.3.2; the classification precision was 70% for the pages and 86% for the ads.

4.1.2 Dataset 2

Dataset 2 is a larger dataset, consisting of 1,756 Web pages, which are also of various types, from online merchant pages to forum pages. After the aforementioned elimination procedure, there remained 827 pages that we used in our experiments. There were 5,065 unique ads and a total of 9,748 judgments.

Table 1 provides average sizes of the individual page fragments defined in Section 3.3.1. The rightmost column shows the number of pages in which each fragment was available. Noteworthy are **M**, **H** and **R**, which were not available for all the pages in both datasets (and hence their overall usefulness should be considered accordingly). The page and referrer URLs (**U** and **R**) were not available for Dataset 2.

4.2 Evaluation Metrics

The standard practice of evaluating IR systems is to perform pooling of judged documents for each query/topic [13]. However, the pooling practice assumes most relevant documents have been judged, and hence considers non-judged documents to be irrelevant. Given the multitude of relevant ads for each page in our case, this solution is inadequate since judged ads constitute only a tiny fraction of all the ads available for retrieval. When each page has numerous relevant ads, it can happen that the top N retrieved ads contain a single judged ad or even none at all. We address this problem in two different ways.

First, Buckley and Voorhees [6] have recently introduced a new evaluation metric, *bpref-10*, which allows to overlook non-judged documents and does not require to consider them to be irrelevant (the metric is computed by analyzing the relative rankings of the relevant and irrelevant documents). To the best of our knowledge, our work is the first study in contextual ad matching that makes use of this new metric in evaluating different matching algorithms.

Second, to compute the standard metrics such as precision or mean average precision (MAP), in our evaluation for each page we consider only those ads for which we have judgments. Each summarization method was applied to this set and the ads were ranked by the score. The relative effectiveness of the methods was determined by comparing how well they separated the ads with positive judgments from those with negative judgments. We present precision at various levels of recall within this set. As the set of judged ads per page is relatively small, this evaluation reports precision that is somewhat higher than it would be with a larger set of negative ads. However, these numbers still establish the relative performance of the algorithms. In Section 4.8 we revisit this issue in greater detail, and for reference conduct an evaluation where we consider non-judged ads to be irrelevant. We demonstrate that in both cases, i.e., whether the non-judged ads are ignored or are considered irrelevant, the performance metrics are highly correlated.

4.3 The Effect of Focused Page Analysis

We now compare the relevance of ad matching when using the entire page vs. the summary of the page.

We examine the performance of different ad matching algorithms that use the following parts of the page:

- Full text (**F**), which embodies all the information that can be gathered from the page per se.
- Full-text + page URL + referrer URL (**F-U-R**), which ads external knowledge from URLs.
- Page title (**T**), which presents a very good balance between text length and informativeness.
- Title, page and referrer URLs, meta data and headings (**U-R-T-M-H**), which combines all the shorter elements of the page.
- Since **U** and **R** components are not available for Dataset 2, we also show for this dataset the performance of the **T-M-A-H-P500** method, which augments the short Title-Meta-Headings summary with anchor text and the first 500 bytes of the page text.

As we can see in Figures 2 and 3, even using the page title alone (**T**) yields matching relevance that is competitive with using all of the page information. The **U-R-T-M-H** method (**T-M-H** for Dataset 2) appears to be the most cost-effective option, as it achieves high relevance scores by analyzing only a few short page excerpts.

4.4 The Contribution of Individual Fragments

Figure 4 shows the contributions of individual page fragments, that is, when the page summary is based on each fragment alone. The fragments are ordered from left to right in the *decreasing* order of their average size (cf. Table 1). Recall that some fragments (notably **M**, **H** and **R**) are available only in some of the pages. Consequently, we evaluated the contribution of each fragment first for all the pages, and then only for pages for which it was available (the corresponding

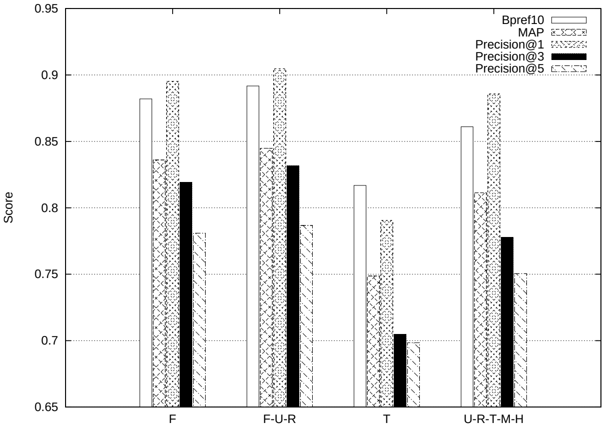

Figure 2: The effect of text summarization (Dataset 1)

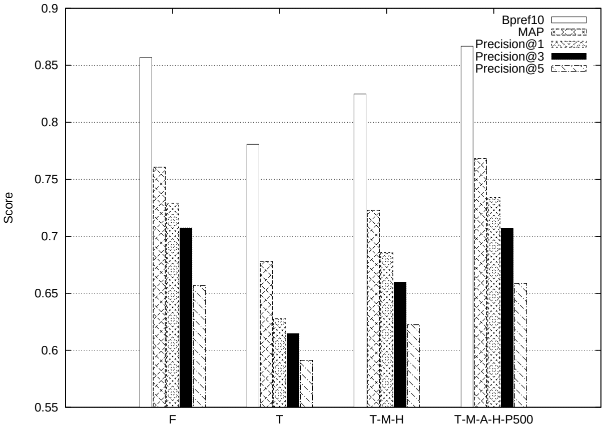

Figure 3: The effect of text summarization (Dataset 2)

graphs are labeled "No zeros" in Figure 4). Predictably, the difference is quite pronounced for **H** and **R**, implying that these components should be used whenever they are available in the page. Figure 5 shows the results for Dataset 2.

The performance of summaries based on the anchor text of outgoing links (**A**) might seem surprising. Intuitively, anchor text characterizes the pages that the current page links to rather than the page itself. However, the anchor text often makes a very good summary of the page itself. For example, a page about high blood pressure might link to pages about heart attacks or medication descriptions that contain relevant information, while pages with lists of items (products, events, etc.) often include links to longer item descriptions. We do not advocate using anchor text in summarization as its size is often quite large (cf. Table 1), but we report this finding because it appeared interesting.

Throughout the paper, we report the results for **P500**, i.e., the initial prefix of the first 500 bytes of the page text. Figure 6 shows the contribution of prefixes of various length.

4.5 Precision-Recall Tradeoff

We show a standard precision-recall graph in Figure 7. Each data point corresponds to the value of precision calculated at a certain percentage of recall. We observe that in all the curves the precision declines gracefully across the entire range of recall levels. We also observe that summaries provide a very good approximation of the full page content over the entire recall range.

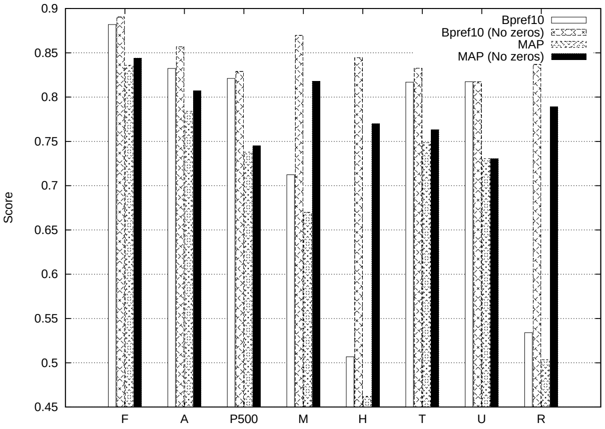

Figure 4: Individual fragments (Dataset 1)

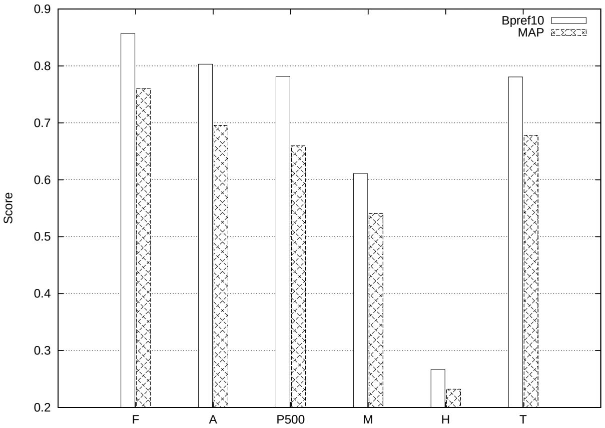

Figure 5: Individual fragments (Dataset 2)

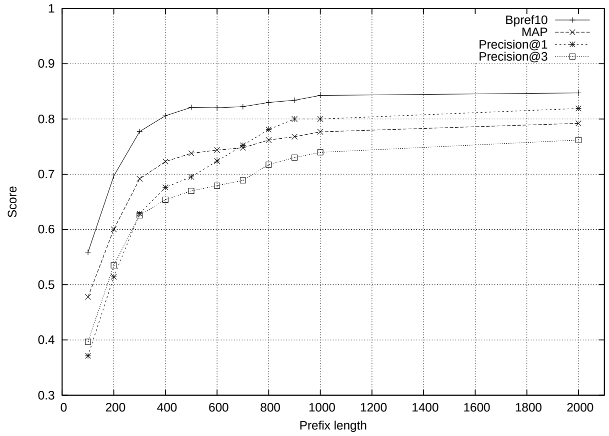

Figure 6: Prefixes of various length (Dataset 1)

4.6 Incremental Addition of Information

Figure 8 plots the performance of increasingly longer summaries, as we progressively incorporate additional page constituents. We add fragments in the increasing order of their length (cf. Table 1). We start with the **U-R** combination, which encompasses external information gathered from the page and referrer URLs, and then add information from the different page parts.

As we can see, even extremely short fragments such as

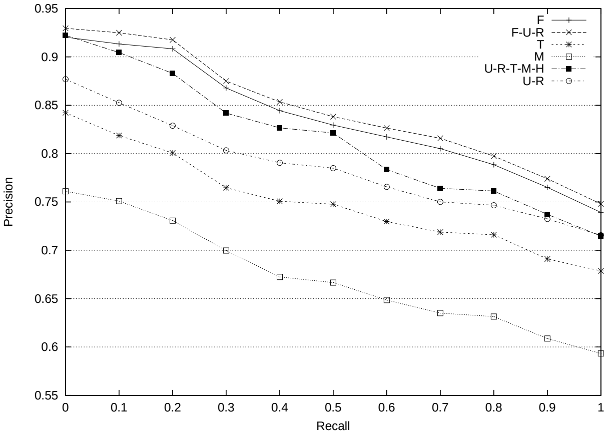

Figure 7: Precision-recall tradeoff (Dataset 1)

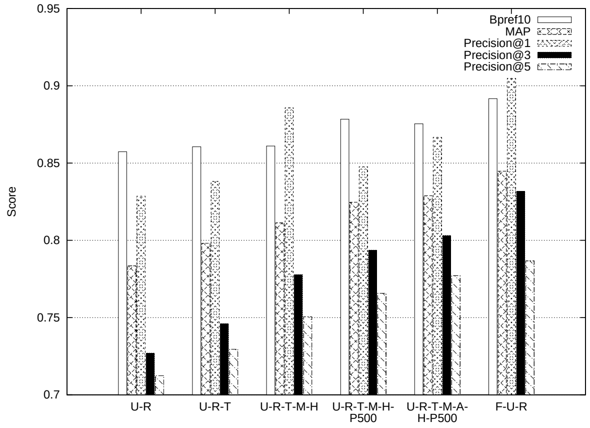

Figure 8: Adding information (Dataset 1)

U-R or **T** carry enough information for successful matching. We also observe that beyond some point using longer summaries becomes unwarranted, as we gain small improvements in relevance in exchange for considerably larger communication and computation load.

4.7 The Effect of Classification

Figure 9 shows the effect of using text classification. We compare ad matching using the following feature sets:

- bag of words (BOW) alone ($\alpha = 1, \beta = 0$)
- classification features alone ($\alpha = 0, \beta = 1$)
- BOW + classification features ($\alpha = 1, \beta = 1$), which is the option used in all other experiments we report.

We observe that the representation based on classification features is surprisingly powerful, and is consistently better than using the words alone. Merging the BOW and the classification features together has a small positive effect, but it might be worth the added complexity, since the number of classification features (5 classes + their ancestors per summary) is much smaller than the BOW.

Previous studies [15, 23] found that text summarization can improve the results of subsequent classification. Although we did not directly evaluate the accuracy of text classification based on summaries, our findings show the benefits of classifying page summaries for ad matching.

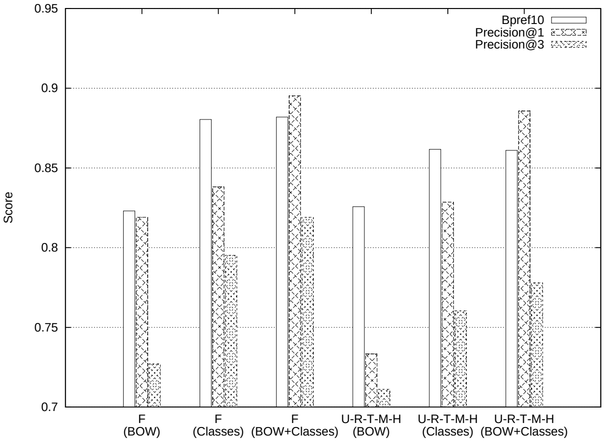

Figure 9: The effect of classification (Dataset 1)

4.8 Considering Non-judged Ads as Irrelevant

The experiments reported above ignored the non-judged ads for each page for the reasons explained in Section 4.2. However, IR practice often considers non-judged documents to be irrelevant, so for the sake of completeness we experimented with this assumption as well. Figure 10 shows the effect of considering non-judged ads as irrelevant. Obviously, the absolute numbers are lower than when non-judged ads are not used. However, the conclusions regarding the utility of text summarization for matching ads still hold.

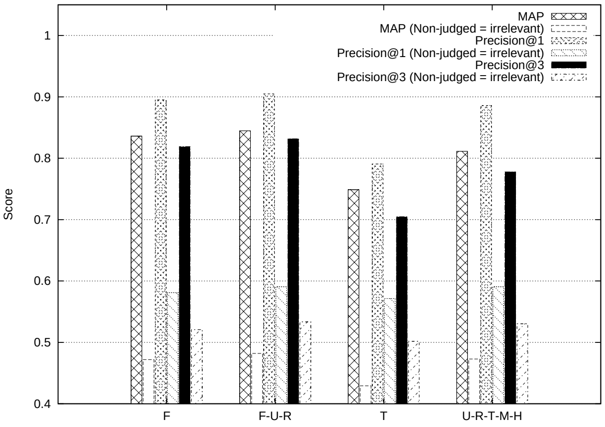

Figure 10: Considering non-judged ads as irrelevant (Dataset 1)

5. RELATED WORK

There are several lines of prior research that are relevant to the work reported herein, including online advertising and text summarization.

5.1 Contextual Ad Matching

Online advertising in general and contextual advertising in particular are emerging areas of research, so the published literature is quite sparse. A recent study [27] confirms the intuition that ads need to be relevant to the user's interest to avoid degrading the user's experience and increase the probability of reaction.

Ribeiro-Neto et al. [19] examined a number of strategies for matching pages to ads based on extracted keywords.

They used the standard vector space model to represent ads and pages, and proposed a number of strategies to improve the matching process. The first five strategies proposed in this work match pages and ads based on the cosine of the angle between their respective vectors. To find the important parts of the ad, the authors explored using different ad sections (e.g., bid phrase, title and body) as a basis for the ad vector. The winning strategy required the bid phrase to appear on the page, and then ranked all such ads by the cosine of the union of all the ad sections and the page vectors. While both pages and ads are mapped to the same space, there is a discrepancy (called "impedance mismatch") between the vocabulary used in the ads and in the pages. For example, the plain vector space model cannot easily account for synonyms, that is, it cannot easily match pages and ads that describe related topics using different vocabularies. The authors achieved improved matching precision by expanding the page vocabulary with terms from similar pages, which were weighted based on their overall similarity to the original page.

In their follow-up work [16], the authors proposed a method to learn the impact of individual features using genetic programming to produce a matching function. The function is represented as a tree composed of arithmetic operators and functions as internal nodes, and different numerical features of the query and ad terms as leaves. The results show that genetic programming finds matching functions that significantly improve the matching compared to the best method (without page-side expansion) reported in [19].

Another approach to contextual advertising is to reduce it to the problem of sponsored-search advertising by extracting phrases from the page and matching them with the bid phrase of the ads. Yih et al. [28] described a system for phrase extraction that uses a variety of features to determine the importance of page phrases for advertising purposes. The system is trained with pages that have been hand-annotated with important phrases. The learning algorithm takes into account features based on *TFIDF*, HTML meta data, and search query logs to detect the most important phrases. During evaluation, each phrase up to length 5 is considered a potential result and evaluated against the trained classifier. In our recent work [5] we experimented with a phrase extractor developed by Stata et al. [24]; however, while slightly increasing the precision, it did not change the relative performance of the explored algorithms.

Langheinrich et al. [17] studied customization techniques for matching ads to users' short-term interests. To capture short-term interests, the authors used search queries as well as visited URLs, which could then be looked up in Web directories.

With the exception of the study by Yih et al. [28], all prior works mostly experimented with the different parts of the ad, assuming the publisher's page is given in its entirety. The latter study did take into account the different page parts (e.g., title, meta data, and specific location of the text on the page), but they used them for a completely different task, namely, identifying good advertising keywords. In contrast, in this work we study the importance of the different parts of the page for the process of contextual ad matching, while our primary aim is to make the matching process as computationally efficient as possible without sacrificing the matching quality.

5.2 Predicting the Clickthrough Rate

An important research direction in web advertising is predicting the clickthrough rate (CTR), that is, the number of clicks a given ad is likely to solicit if displayed on a given page.

Regelson and Fain [18] estimated the CTR by clustering ads by their bid phrases. The clickthrough rate was averaged over each cluster, and the CTR estimate for new ads was obtained by finding the nearest cluster. More recently, Richardson et al. [20] estimated the clickthrough rate by analyzing the different parts of the ads (e.g., bid phrases, landing page, and title). Again, both works focused on the ad side of the matching problem, while we study the role of the different parts of the page to which ads are matched.

5.3 Web Page Summarization

Our analysis of parts of the page instead of the entire page for ad matching relies on the findings of prior studies in Web page summarization. The latter is different from general text summarization in two important aspects. First, it relies on markup and other clues that are typically found on Web pages but not in plain text documents. Second, Web pages are often more noisy and generally do not qualify as Standard Written English, which is often assumed in mainstream text summarization.

Buyukkokten et al. [7], and later Alam et al. [1] studied summarization of Web pages for presentation on handheld devices. Sun et al. [25] summarized Web pages by using clickthrough data from a search engine, which allowed them to associate pages with queries that retrieved them. The authors argued that when users click on a search result retrieved for a given query, the words of a query can be viewed as highly characteristic of the page content, and thus useful in its summary. Jatowt and Ishizuka studied the effect of the dynamic nature of Web pages on their summarization [14]. The authors proposed to collectively analyze historic versions of the page to gain insights into the terms that are most characteristic of this page. Berger and Mittal [2] argued that Web pages often lack coherent text and well-defined discourse structure, and consequently extractive summarization techniques are not applicable to them. To address the peculiar nature of Web page summarization, they proposed to perform non-extractive summarization by "translating" a page using techniques based on statistical machine translation.

Several works studied the synergy between text summarization and text classification. Kolcz et al. [15] used summaries to perform feature selection, assuming that terms that occur in the summary are more informative for categorization. Shen et al. [23] also found that carefully crafted summaries of pages can notably increase the precision of text classification by eliminating less important and more noisy parts of the page. Both these works found that page title, first paragraph and meta fields (keywords/description) carry a significant amount of information about the page.

6. CONCLUSIONS AND FUTURE WORK

We presented a new methodology for contextual Web advertising in real time. Prior works in the field explored the relative importance of the different constituent part of ads. In this work, we focused on the contributions of the different fragments of the pages. Extracting small but informative parts of pages is important because often page content is not available for analysis ahead of time, as is the case for dynamically created or frequently updated pages.

Our approach allows to match ads to pages in real time, without prior analysis of the page content. Our solution is easy to implement within the standard JavaScript mechanisms used for ad placement, and adds only 500–600 bytes to the usual request for ads. We employ text summarization techniques to identify short but informative page fragments that can serve as a good proxy for the entire page. We also use two source of external knowledge. First, we extract information from the page and referrer URLs, which often contain words pertinent to the page topic. Second, we use text classification techniques to classify the page summary with respect to a large taxonomy of commercial topics.

Experimental findings confirm that using only a small portion of the page text can yield highly relevant ads, and the quality of summary-based ad matching is competitive with that of using the full page. For example, for Dataset 1 we observed that using only 5% of the page text can still yield 97%–99% of the full-text-based relevance (94%–99% for Dataset 2). We identified the various key parts of the page, and analyzed their contributions collectively and individually. Our results also confirmed that page-ad matching can be improved by classifying page summaries, and matching pages and ads in the augmented space of words and classification-based features.

In our experiments, we observed that in some cases merely taking the first few hundred bytes of the page text also yields reasonable results. However, using the page prefix rather than the page structure entails some caveats: it raises higher privacy concerns (if the page is personalized) and it is easier to spam. Further observation and experimentation is necessary, in particular for long pages. In future work, we also plan to experiment with different weighting of the various page fragments, using machine learning techniques to determine the optimal weights. We also plan to examine ways of constructing the summary based on the page type (e.g., for a blog page, the prefix information might be useful as it is likely to contain the most recent postings, while for a concert listing, the anchor text might be of crucial importance).

Acknowledgments

We thank our colleagues Bo Pang for the text tokenization module, Prashanth Bhat for the ad indexer, and Kishore Papineni for suggesting the idea of extracting words from URLs. We also thank Arkady Estrin for JavaScript advice.

7. REFERENCES

[1] Hassan Alam, Rachmat Hartono, Aman Kumar, Fuad Rahman, Yuliya Tarnikova, and Che Wilcox. Web page summarization for handheld devices: A natural language approach. In *ICDAR'03*, 2003.

[2] Adam Berger and Vibhu O. Mittal. OCELOT: a system for summarizing web pages. In *SIGIR'00*, 2000.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW'98*, 1998.

[4] Andrei Broder, Marcus Fontoura, Evgeniy Gabrilovich, Amruta Joshi, Vanja Josifovski, and Tong Zhang. Robust classification of rare queries using web knowledge. In *Proceedings of the 30th ACM International Conference on Research and Development in Information Retrieval*, 2007.

[5] Andrei Broder, Marcus Fontoura, Vanja Josifovski, and Lance Riedel. A semantic approach to contextual advertising. In *SIGIR'07*. ACM Press, 2007.

[6] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR'04*, 2004.

[7] Orkut Buyukkokten, Oliver Kaljuvee, Hector Garcia-Molina, Andreas Paepcke, and Terry Winograd. Efficient web browsing on handheld devices using page and form summarization. *ACM Transactions on Information Systems*, 20(1):82–115, January 2002.

[8] P. Chatterjee, D. L. Hoffman, and T. P. Novak. Modeling the clickstream: Implications for web-based advertising efforts. *Marketing Science*, 22(4):520–541, 2003.

[9] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.

[10] D. Fain and J. Pedersen. Sponsored search: A brief history. In *Second Workshop on Sponsored Search Auctions*, 2006.

[11] Evgeniy Gabrilovich and Shaul Markovitch. Feature generation for text categorization using world knowledge. In *IJCAI'05*, pages 1048–1053, 2005.

[12] Eui-Hong (Sam) Han and George Karypis. Centroid-based document classification: Analysis and experimental results. In *PKDD'00*, September 2000.

[13] D. Hawking, N. Craswell, and P.B. Thistlewaite. Overview of TREC-7 very large collection track. In *TREC-7*, 1998.

[14] Adam Jatowt and Mitsuru Ishizuka. Web page summarization using dynamic content. In *WWW'04*, 2004.

[15] Aleksander Kolcz, Vidya Prabakarmuthi, and Jugal Kalita. Summarization as feature selection for text categorization. In *SIGIR'01*, pages 365–370, 2001.

[16] Anisio Lacerda, Marco Cristo, Marcos Andre Goncalves, Weiguo Fan, Nivio Ziviani, and Berthier Ribeiro-Neto. Learning to advertise. In *SIGIR'06*, pages 549–556, 2006.

[17] Marc Langheinrich, Atsuyoshi Nakamura, Naoki Abe, Tomonari Kamba, and Yoshiyuki Koseki. Unintrusive customization techniques for web advertising. *Computer Networks*, 31:1259–1272, May 1999.

[18] M. Regelson and D. Fain. Predicting click-through rate using keyword clusters. In *Second Workshop on Sponsored Search Auctions*, 2006.

[19] Berthier Ribeiro-Neto, Marco Cristo, Paulo B. Golgher, and Edleno Silva de Moura. Impedance coupling in content-targeted advertising. In *SIGIR'05*, 2005.

[20] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: Estimating the click-through rate for new ads. In *WWW'07*. ACM Press, 2007.

[21] J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.

[22] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[23] Dou Shen, Zheng Chen, Hua-Jun Zeng, Benyu Zhang, Qiang Yang, Wei-Ying Ma, and Yuchang Lu. Web-page classification through summarization. In *SIGIR'04*, 2004.

[24] Raymond Stata, Krishna Bharat, and Farzin Maghoul. The term vector database: fast access to indexing terms for web pages. *Computer Networks*, 33(1–6):247–255, 2000.

[25] Jian-Tao Sun, Dou Shen, Hua-Jun Zeng, Qiang Yang, and Zheng Lu, Yuchang anf Chen. Web-page summarization using clickthrough data. In *SIGIR'05*, pages 194–201, 2005.

[26] W3C. Document Object Model, Level 1 Specification, 2005.

[27] C. Wang, P. Zhang, R. Choi, and M. D. Eredita. Understanding consumers attitude toward advertising. In *8th Americas Conference on Information Systems*, 2002.

[28] Wen-tau Yih, Joshua Goodman, and Vitor R. Carvalho. Finding advertising keywords on web pages. In *WWW'06*, 2006.