Web Page Summarization for Just-in-Time Contextual Advertising

ARIS ANAGNOSTOPOULOS, Sapienza University of Rome
ANDREI Z. BRODER, EVGENIY GABRILOVICH, VANJA JOSIFOVSKI,
and LANCE RIEDEL, Yahoo! Research

Contextual advertising is a type of Web advertising, which, given the URL of a Web page, aims to embed into the page 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, often ads need to be matched to new or dynamically created pages that cannot be processed ahead of time. Analyzing the entire content 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 6% fraction of the page text can sacrifice 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. We also compared our summarization approach, which is based on structural properties of the HTML content of the page, with a more principled one based on one of the standard text summarization tools (MEAD), and found their performance to be comparable.

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

Additional Key Words and Phrases: Text classification, text summarization

ACM Reference Format:
Anagnostopoulos, A., Broder, A. Z., Gabrilovich, E., Josifovski, V., and Riedel, L. 2011. Web page summarization for just-in-time contextual advertising. ACM Trans. Intell. Syst. Technol. 3, 1, Article 14 (October 2011), 32 pages.
DOI = 10.1145/2036264.2036278 http://doi.acm.org/10.1145/2036264.2036278

14

1. INTRODUCTION

A recent IDC report estimates that the total Internet advertiser spending in 2009 at 61 billion dollars (26 for U.S. only), and predicts an annual growth rate of 16% over the next 5 years. 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

A preliminary version of this article appeared in *Proceedings of the 16th ACM Conference on Information and Knowledge Management* [Anagnostopoulos et al. 2007].
Authors' addresses: A. Anagnostopoulos (corresponding author), Department of Informatics and System Sciences, Sapienza University of Rome, Via Ariosto 25, 00183 Rome, Italy; email: aris@dis.uniromal.it; A. Z. Broder, E. Gabrilovich, V. Josifovski, and L. Riedel, Yahoo! Research, 4301 Great America Parkway, Santa Clara, CA 95054.

DOI 10.1145/2036264.2036278 http://doi.acm.org/10.1145/2036264.2036278

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 article, 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 of 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 servings 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 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. Note that the JavaScript code is embedded in the page at the location where ads are to be displayed.
(2) In line with previous work on full pages [Broder et al. 2007b], 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 to recrawl 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 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 article are fourfold.

— 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.
— Finally, we compare two different approaches to page summarization: the proposed approach based on structural properties of the HTML content of the page, and a more principled one based on one of the standard summarization tools available (MEAD) [Radev et al. 2003]. Our findings suggest that for this particular application the performance of the two approaches is comparable, while the former one is much more efficient and can actually be used "just in time" at page display time.

The rest of the article is organized as follows. Section 2 provides background on current practices in Web advertising. Section 3 presents 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.

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 Fain and Pedersen [2006] for a brief history of the subject). Today, almost all of the for-profit nontransactional 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.

[1]Nontransactional sites are those that do not sell anything directly.

— 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 confirm that improved relevance increases the number of ad-clicks [Chatterjee et al. 2003; Wang et al. 2002].

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. For completeness we next describe the PPC model, although our methodology is independent of the pricing model.

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 [Ribeiro-Neto et al. 2005]. 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

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

where k is the number of ads displayed on page p and $price(a_i, i, p, \mathcal{A})$ is the click-price of the given ad a_i at position i, at page p, when the rest of the ads (and their placement)

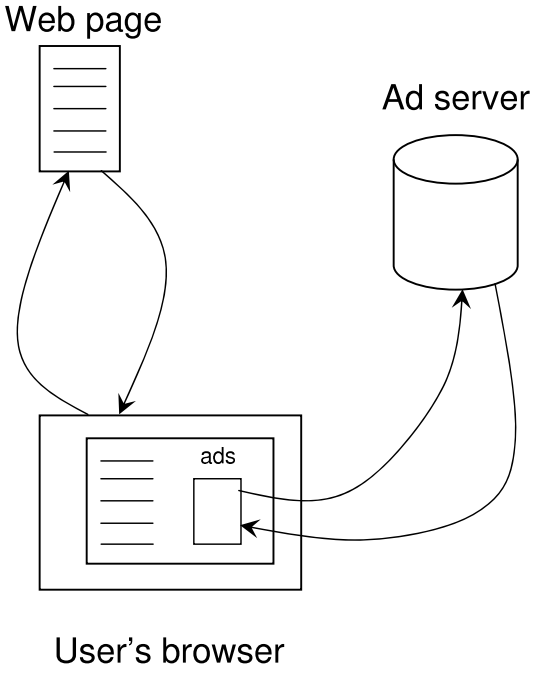

Fig. 1. Overview of ad display.

is described by $\mathcal{A}$. 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 (e.g., Edelman et al. [2007]). In this article, we do not address the pricing model and we concentrate on finding ads that will maximize the first term of the product, that is, we search for

$$\arg\max_i P(click|p, a_i, \mathcal{A}).$$

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, and the effect of the rest of the ads shown. We assume that these are orthogonal factors to the relevance component and could be incorporated into 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.

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 precrawl 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 precrawling 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 versus analyzing as little as possible to save transmission and analysis time.

To produce summaries, we use clues from the HTML structure of the Web page. To this end, we employ a number of techniques [Buyukkokten et al. 2002; Kolcz et al. 2001] 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 as follows.

(1) *URL.* We tokenize (in the server) 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 (in the server) 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 in the server the page content onto a large taxonomy and use resultant classifications to augment page representation for ad matching. To this end, we preclassify 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 before. However, it was previously found [Kolcz et al. 2001; Shen et al. 2004] 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 [Brin and Page 1998]. 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. Web-Page-Aware Text Summarization. Text summarization techniques are divided into *extractive* and *nonextractive* 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 extremely 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 [Buyukkokten et al. 2002; Kolcz et al. 2001], we evaluate the role of the following page components in constructing summaries:

— Title (**T**)
— Meta keywords and description (**M**)

[2]Document Object Model (DOM) is a standard approach to representing HTML/XML documents [W3C 2005].

— 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**)

We note here that all these methods incur minimal impact on the client side in the range of a few milliseconds. In the next section, we evaluate the individual contribution of each of the aforelisted page segments as well as their combinations for serving a page proxy for ad matching.

URL tokenization was performed on the server, and hence incurred no client-side overhead. To tokenize URLs into words, we used a dynamic programming tool developed in-house, which relied on a unigram language model built from a corpus of several million ad documents. More specifically, consider a string with n characters $c_1, c_2, ..., c_n$. Let D denote a dictionary that assigns a log probability value to each word in the corpus. We want to obtain an optimum set of segmentation points according to the dictionary. Let $c[i..j]$ denote the substring from the i-th character to the j-th character in the string, we define the subproblem for the substring $c[1..i]$ as

$$s(i) = max_{k\in[1..i]} s(k) + score(k+1, i),$$

where

$$score(i, j) = \begin{cases} logprob(c[i..j]), & \text{if } c[i..j] \in \text{D} \\ logprob(w_{unknown}) * (j + 1 - i), & \text{otherwise.} \end{cases}$$

We set $logprob(w_{unknown})$ to be lower than the log probability of any word observed in the corpus, and by lowering the score for an unknown subsequence of the string proportional to its length, shorter unknown sequences are preferred over longer ones. Backtracking the set of k that yields the best $s(n)$ gives us the optimal set of segmentation points for the string. In our tests, the tokenization code proved very efficient, taking approximately 0.25 msec per URL (averaged over 160,000 real-world URLs).

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 effects 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 [Kolcz et al. 2001; Shen et al. 2004]. A recent study also found that features constructed with the aid of a knowledge-based taxonomy are beneficial for text classification [Gabrilovich and Markovitch 2005]. 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 pages into one node will likely result in poor ad matching, as both "sore foot" and "flu" pages will end up in the same node. The ads appropriate for these two pages are, however, very different. To avoid such situations, the taxonomy needs

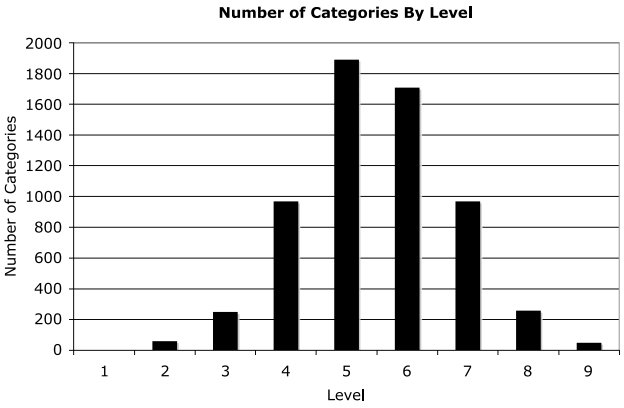

Fig. 2. Taxonomy statistics: number of categories per level.

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 (approximately 150 phrases per node), which were used as a training set; a small fraction of queries have been assigned to more than one category. Ideally, it is, of course, preferable to have labeled training documents from the same distribution from which documents to be classified are drawn. Since our method classifies Web search results, labeled training examples should ideally also be Web pages. It is, however, prohibitively expensive to manually label a large enough set of Web pages at the resolution we need (i.e., to populate a taxonomy of 6,000 nodes). It is substantially less expensive to label short bid phrases rather than long documents. Using labeled bid phrases of ads is also particularly suitable for our application, since our research is motivated by the need to match queries to more relevant ads.

The taxonomy has been populated by human editors using keyword suggestions tools similar to the ones used by ad networks to suggest keywords to advertisers. After initial seeding with a few queries, using the provided tools a human editor can add several hundreds queries to a given node, which were used as a training set. A small fraction of queries have been assigned to more than one category. Some queries were assigned to more than one category because they had several equally important facets. For example, a query about antivirus software for Linux could be simultaneously assigned to categories *"Computing/Computer Security/Malicious Software Prevention and Elimination/Virus Utilities/Anti Virus Utilities - Linux"* and *"Computing/Computer Software/Software Utilities/Security Software/Firewalls/Firewalls - Linux"*. Here, the former classification emphasizes the security application, and the latter the fact that the application is implemented in software rather than in hardware. Nevertheless, it has been a significant effort to develop a taxonomy of a magnitude of several person-years. A similar-in-spirit process for building enterprise taxonomies via queries has been presented by Gates et al. [2005]. However, the details and tools are completely different. Figures 2 and 3 show pertinent statistics about the structure of the taxonomy, and Figures 4 and 5 show statistics about the labeled examples used to train the classifier described in Section 3.3.2.

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 [Duda

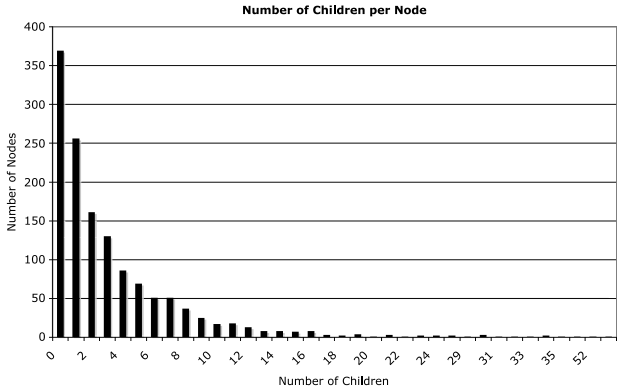

Fig. 3. Taxonomy statistics: fanout of nonleaf nodes.

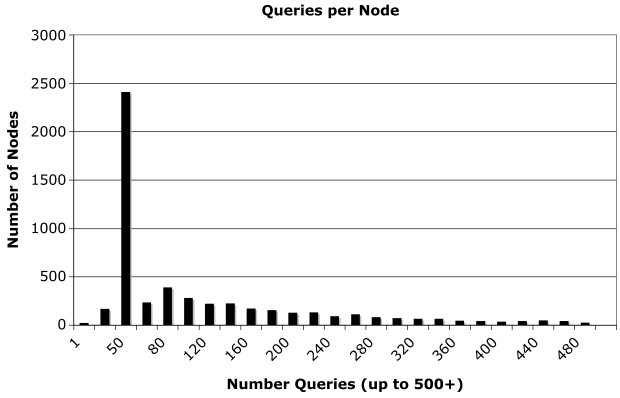

Fig. 4. Taxonomy statistics: number of training examples (queries) per node.

and Hart 1973], as well as prototype formation methods such as Rocchio [Rocchio 1971] or centroid-based [Han and Karypis 2000] 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 metadocument. 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 metadocuments. We have

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

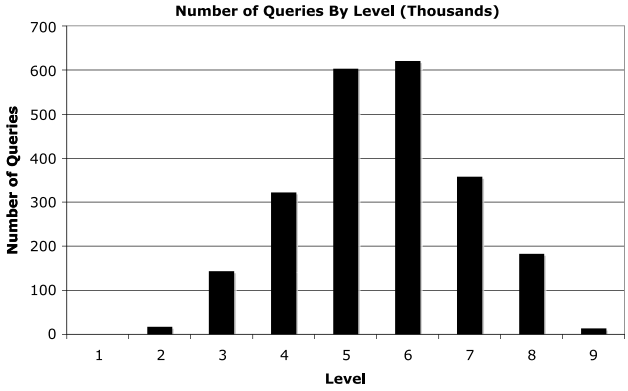

Fig. 5. Taxonomy statistics: number of training examples (queries) per level.

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 [Salton and Buckley 1988].

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 Gabrilovich and Markovitch [2005], 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.

3.3.3. Classical Text Summarization. The preceding approach to Web page summarization uses structural clues based on the HTML structure of the Web page, and is therefore very efficient. However, it would be interesting to determine whether more elaborate (and correspondingly more computationally heavy) techniques could produce better summaries for ad placement. To address this question, we also experimented with MEAD (in particular with the latest version at the time we performed the experiments, version 3.10), one of the standard text summarization tools [Radev et al. 2003]. In what follows, we briefly describe how the basic functionality of MEAD works. Later, in Section 4.9, we report the results obtained with MEAD. Note that our goal is not to use the most advanced text summarization techniques but instead to examine how do carefully selected Web page field extracts comparison to more advanced text summarization technology (which, however, cannot be used for our application due to time constraints). As we see in our experiments, the two techniques have comparable performance.

After parsing each Web page and extracting the text, MEAD processes each sentence. In particular, the MEAD classifier scores each sentence according to features such as its length, or similarity with the rest of the document. After the sentences are

scored, a reranker reorders them based on their score and the similarity to the higher ranked sentences. Finally, based on the scores, the desired number of sentences is extracted and included in the summary.

This is the default execution of the MEAD package. There is some additional work implemented in MEAD for summarization, most notable the LexRank feature [Erkan and Radev 2004]. We did not use the LexRank feature as it is mostly suitable for summarizing large document collections to provide meaningful results, while in our case we desire summaries of small individual Web pages.

3.4. Efficiency Considerations

As we explained before, there are numerous scenarios that do not allow ahead-of-time analysis of Web pages for ad placement. These scenarios include frequently changing pages, pages generated on-the-fly or personalized for the user, as well as pages that are not accessible ahead of time without user authentication. Therefore, to present users with relevant ads, the only feasible option is to perform fast, client-side summarization of the Web page, and then send the resultant short summary to the server for analysis and ad matching.

Traditional text summarization uses complex models, which are computationally and resource-intensive, and hence cannot be possibly deployed on the client side, let alone in the standard browser application. Traditional text summarization can only be done on the server side, which requires the entire (and often very long) Web page to be transmitted to the server.

In contrast, our Javascript-based method performs summarization nearly instantaneously. Observe that in order to render the Web page and show it to the user, the browser parses the HTML content of the page anyway. Consequently, JavaScript can access the DOM tree of the HTML document to fetch the needed page parts (headings, metakeywords, etc.) with essentially zero overhead [Powers 2008]. Furthermore, as we discuss in Section 4 shortly, in most cases the client-based summary produced by concatenating the carefully identified parts of the page is only about 500 bytes long. Compared to a typical Web page having 10 kilobytes of text (and often much more), this is a 20-fold reduction in the amount of information to be transferred to the server. We also believe that the time required to process that information for ad matching is also reduced proportionally. Therefore, the key advantages of our approach are: (a) the possibility to create a short summary on the client side with essentially zero overhead, and (b) saving the transmission and subsequent processing times many-fold by drastically (yet carefully) reducing the amount of information.

For very large documents the computational overhead at the server for complex summarization can become significant. The additional time depends of course on the particular technique. For example, for the MEAD tool used here, in the worst case it is $\Omega(\ell \ln \ell + \ell s)$, where ℓ is the number of sentences in the document and s is the desired number of sentences in the summary. This is because MEAD, after scoring the document sentences, sorts them and then, before inserting a candidate sentence in the summary, compares it with all the sentences already selected for inclusion. Nevertheless, as we mentioned, while for large documents this time might be nonnegligible, we believe that the main bottleneck is the transfer time.

As we show soon, our approach is not only very efficient, but is also highly effective. Specifically, we show that the summary we produce does not sacrifice the relevance of ads matched to it, compared to matching the ads to the entire (i.e., not summarized) Web page. We further show that using classical text summarization techniques would not lead to improved relevance of ads (even though performing such summarization on the client side is not feasible at all).

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 that follow 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 that participated in the Yahoo!'s Content Match product in 2007. Ads for each of these pages have been selected from a large pool of about 30 million ads in the system at that time. We preprocessed both pages and ads by removing stop words 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 an NFL page.

We note here that internal studies of the relationship between the relevance scores and the click-through rate of the contextual ads have shown significant correlation.

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 shortly 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. Since the search engines use site reputation and content-based metrics among other factors in the ranking, the pages in this dataset tend to be of better quality with more textual content, as confirmed by visual evaluation. 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 I and visually in Figures 6 and 7. The pages 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. Also, due to the way the corpus was composed, the pages in Dataset 1 have on average twice as many judgments as in Dataset 2 (28 versus

Table I. Sizes of Page Fragments

Page fragment		Dataset 1		Dataset 2	
Description	Short-hand	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	–	–

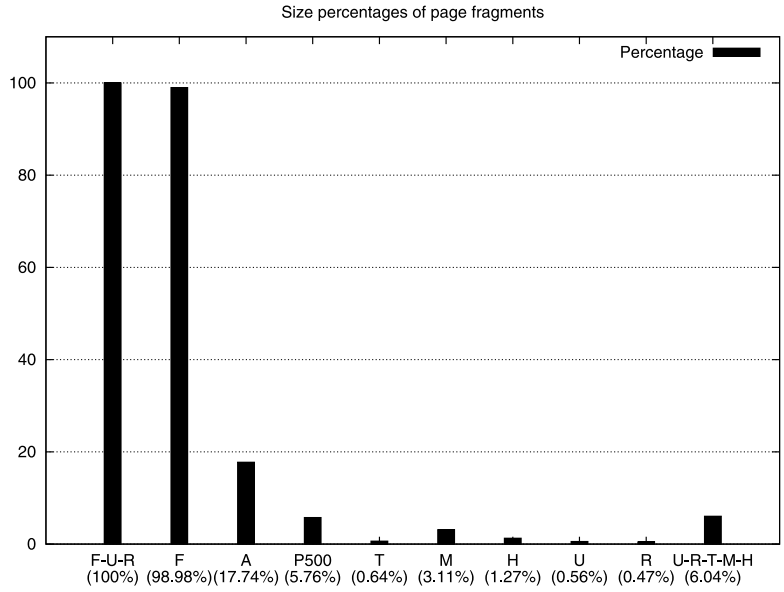

Fig. 6. Size percentages of various segments and their combinations for Dataset 1. F corresponds to the full page, U to the page's URL, R to the referrer page's URL M to the metadata, H to the headings sections of the page, A to the anchor text of the page's outgoing links, and P500 to the page's first 500 bytes.

11.7). For these reasons we emphasize Dataset 1 in our evaluation, while presenting results for Dataset 2 as an approximate indication of how our technique would apply to lower-quality pages.

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 before), 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). Interjudge agreement in scoring was 84%. We

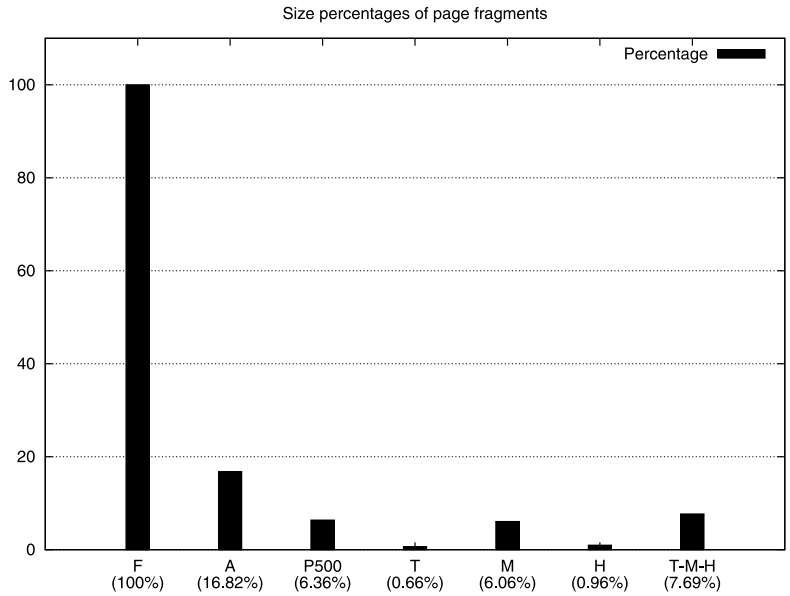

Fig. 7. Size percentages of various segments and their combinations for Dataset 2. F corresponds to the full page, M to the metadata, H to the headings sections of the page, A to the anchor text of the page's outgoing links, and P500 to the page's first 500 bytes.

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 I 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, since when the data was collected they were not stored.

4.2. Evaluation Metrics

The standard practice of evaluating IR systems is to perform pooling of judged documents for each query/topic [Hawking et al. 1998]. However, the pooling practice assumes most relevant documents have been judged, and hence considers nonjudged 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 [2004] have recently introduced a new evaluation metric, *bpref-10*, which allows to overlook nonjudged documents and does not require to consider them 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. The bpref-10 measure is defined as

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

where R is the number of relevant documents, r is a relevant document (so the summation is over all the relevant documents), and n is a member of the top $10+R$ nonrelevant documents.

Second, we use some standard metrics, precision at k:

$$\text{precision-at-}k = \frac{\#\text{ relevant docs in top-}k}{k},$$

and mean average precision (MAP):

$$\text{MAP} = \frac{\sum_{k=1}^{N} \text{precision-at-}k \cdot \mathbf{1}_{\text{document at rank } k \text{ is relevant}}}{\#\text{ relevant docs}},$$

where N is the number of all the documents and $\mathbf{1}_A$ is the indicator function for predicate A. In other words, MAP is the average precision over all recall levels. However, in our evaluation with these two metrics, 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 have been 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 nonjudged ads to be irrelevant. We demonstrate that in both cases, that is, whether the nonjudged ads are ignored or are considered irrelevant, the performance metrics are highly correlated[3] and, hence, the conclusions that we draw in either case are the same.

Overall we report values for bpref-10 and for precision at 1, 3, and 5, while in some more loaded graphs we might drop some of these metrics to reduce clutter. The results that we have obtained with the missing values, however, are consistent with those that we report.

4.3. The Effect of Focused Page Analysis

We now compare the relevance of ad matching when using the entire page versus 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. This forms our baseline that we compare the summarization techniques with.

[3]That is, for every metric (MAP, P@1, P@3), the relative order of the scores computed using various page fragment combinations is the same for both methods. See, for example, Figures 8 and 16.

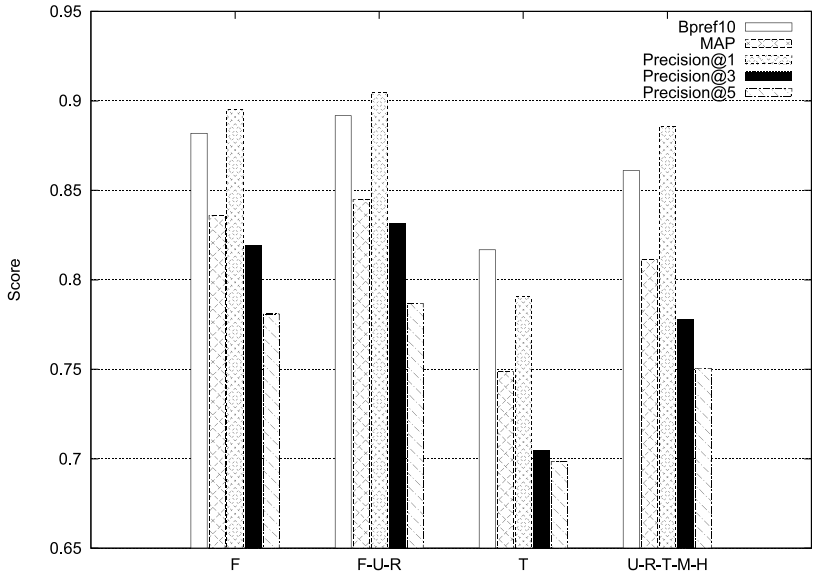

Fig. 8. The scores of various text summarization options for Dataset 1. F corresponds to the full page, U to the page's URL, R to the referrer page's URL, M to the metadata, and H to the headings sections of the page. The results are with classification.

— Page title (**T**), which presents a very good balance between text length and informativeness.
— Title, page, and referrer URLs, metadata, 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 8 and 9, 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 for large pages, as it achieves high relevance scores by analyzing only a few short page excerpts. In particular, using only 6% of the Web page content for Dataset 1 and 7.7% for Dataset 2, it is able to achieve a relevance of 97%–99% for Dataset 1 and 94%–99% for Dataset 2. The difference in the two datasets indicates also that Web page summarization is more cost effective for large pages; for shorter pages it becomes less efficient and for very short pages (with length below a certain threshold) a real system deployment might be more effective by using the entire Web page information.

4.4. The Contribution of Individual Fragments

Figure 10 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 (refer to Table I). 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 graphs are labeled "No zeros" in Figure 10). Predictably, the difference is quite pronounced for **H** and **R**, implying that

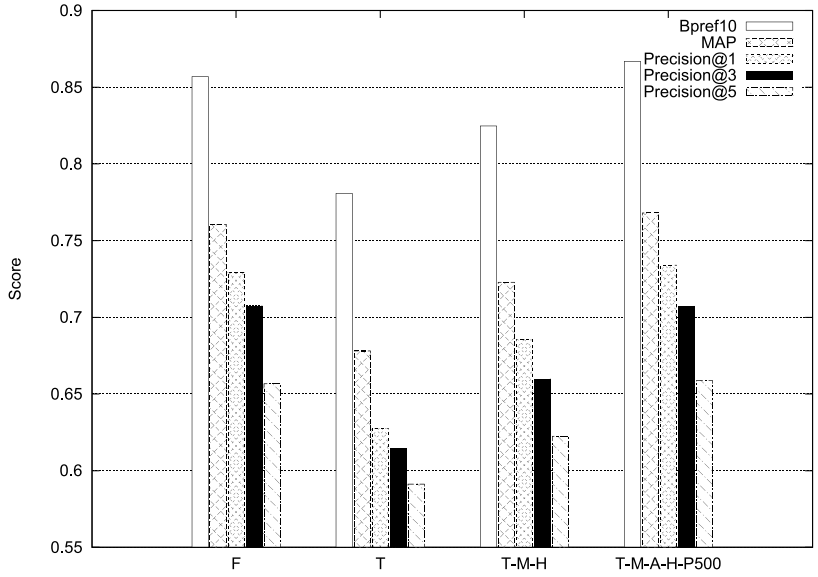

Fig. 9. The scores of various text summarization options for Dataset 1. F corresponds to the full page, M to the metadata, to H the headings sections of the page, A to the anchor text of the page's outgoing links, and P500 to the page's first 500 bytes. The results are with classification.

these components should be used whenever they are available in the page. Figure 11 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 (refer to Table I), but we report this finding because it appeared interesting.

Throughout the article, we report the results for P500, that is, the initial prefix of the first 500 bytes of the page text. Figure 12 shows the contribution of prefixes of various length for Dataset 1.

4.5. Precision-Recall Trade-Off

We show a standard precision-recall graph in Figure 13. 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.

4.6. Incremental Addition of Information

Figure 14 plots the performance of increasingly longer summaries, as we progressively incorporate additional page constituents. We add fragments in the increasing order of their length (refer to Table I). 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.

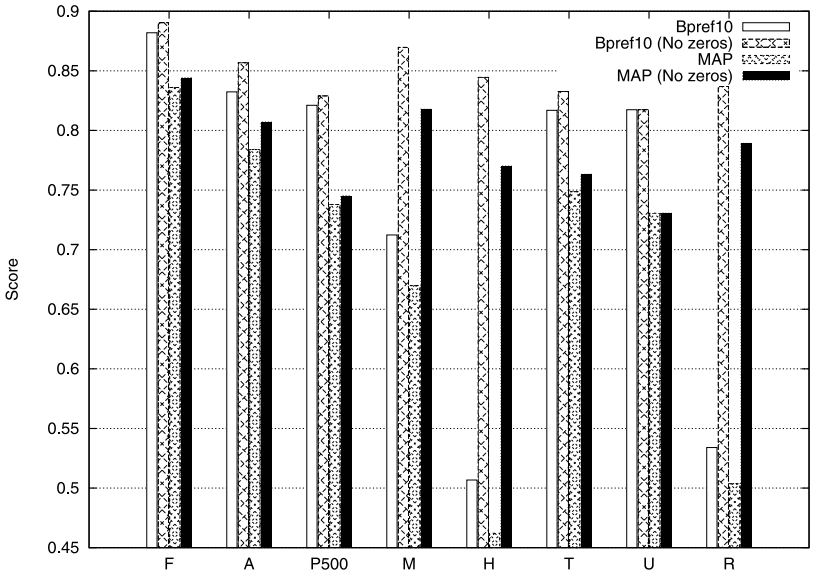

Fig. 10. The scores of individual page fragments for Dataset 1. The values in the x-axis are sorted in decreasing order of average length. F corresponds to the full page, A to the anchor text of the page's outgoing links, P500 to the page's first 500 bytes, M to the metadata, H the headings sections, T to the page's title, U to the page's URL, and R to the referrer page's URL. The results are with classification. The two sets of scores differ by the way they average: Bpref10 and MAP show the average score over all the documents; Bpref10 (No zeros) and MAP (No zeros) show the average score over the documents for which the corresponding page fragment exists.

As we can see, even extremely short fragments such as U-R 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 15 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 the 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 [Kolcz et al. 2001; Shen et al. 2004] 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.

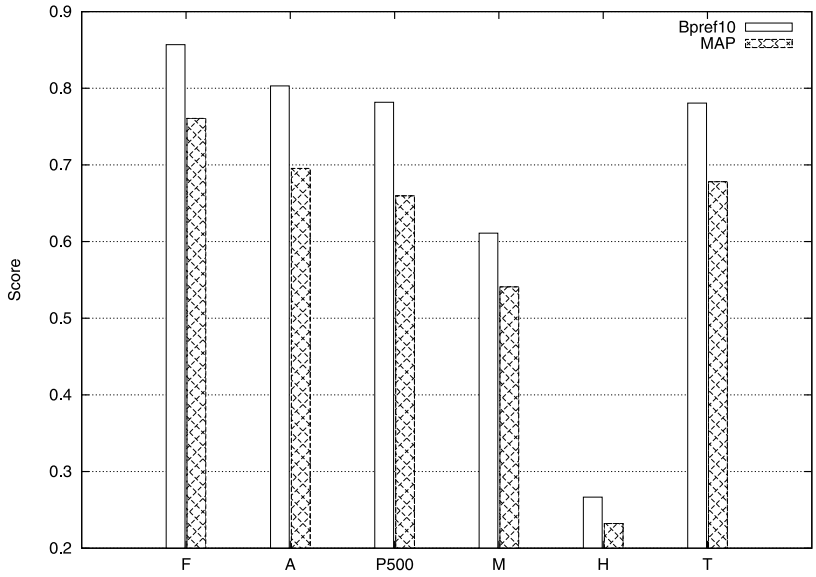

Fig. 11. The scores of individual page fragments for Dataset 2. The values in the x-axis are sorted in decreasing order of average length. F corresponds to the full page, A to the anchor text of the page's outgoing links, P500 to the page's first 500 bytes, M to the metadata, H the headings sections, and T to the page's title. The results are with classification.

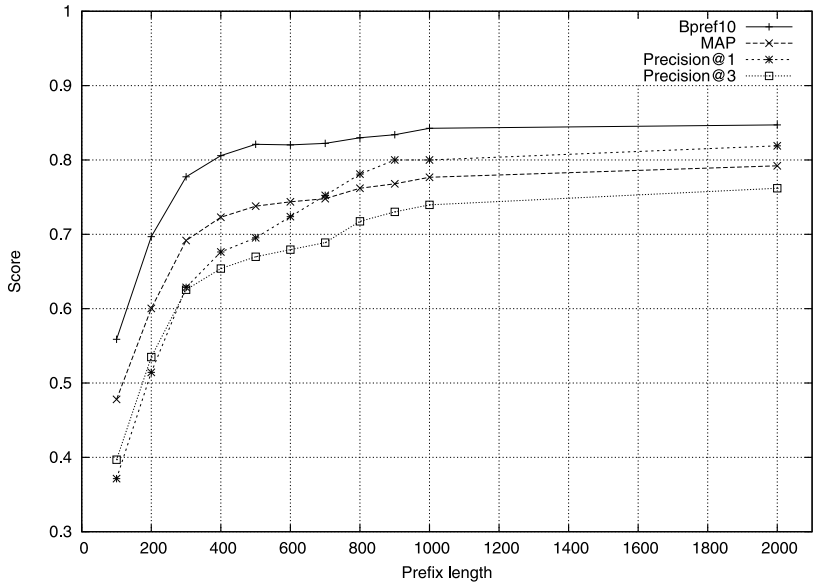

Fig. 12. The scores with prefixes of various length for Dataset 1. The values in the x-axis (denoted with Px in the text) correspond to the length in bytes of the prefix used. The results are with classification.

4.8. Considering Nonjudged Ads as Irrelevant

The experiments reported earlier ignored the nonjudged ads for each page for the reasons explained in Section 4.2. However, IR practice often considers nonjudged documents to be irrelevant, so for the sake of completeness we experimented with this

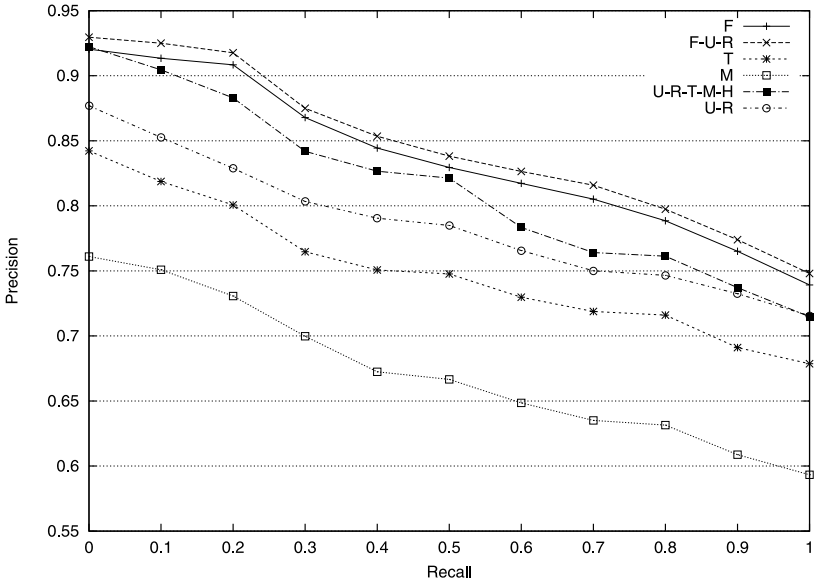

Fig. 13. Precision-recall trade-off for Dataset 1. F corresponds to the full page, U to the page's URL, R to the referrer page's URL, T to the page's title, M to the metadata, and H to the headings sections of the page. The results are with classification.

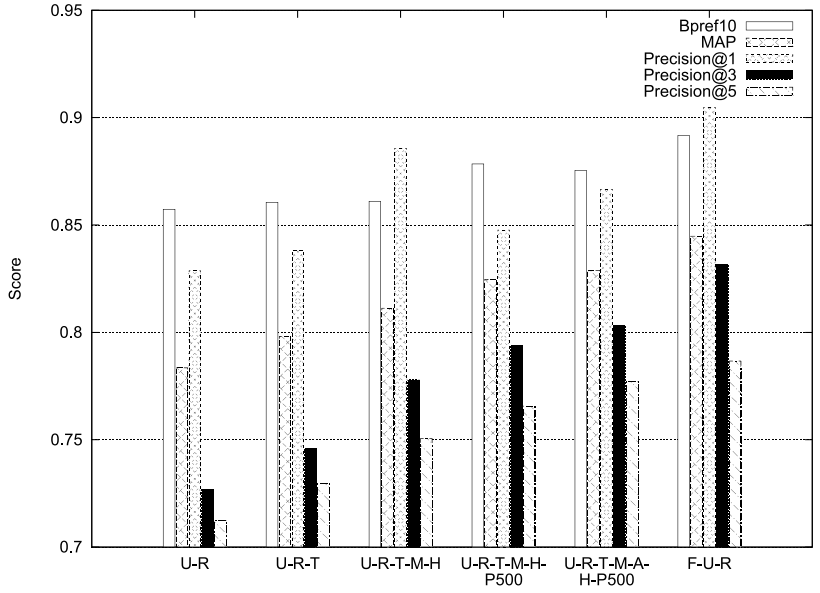

Fig. 14. Score when we incrementally add information in Dataset 1. U corresponds to the page's URL, R to the referrer page's URL, T to the page's title, M to the metadata, H to the headings sections, P500 to the first 500 bytes, A to the anchor text of the outgoing links of the page, and F to the full page. The results are with classification.

assumption as well. Figure 16 shows the effect of considering nonjudged ads as irrelevant. Obviously, the absolute numbers are lower than when nonjudged ads are not used. However, we can see that we obtain the same conclusions, that is, the relevant

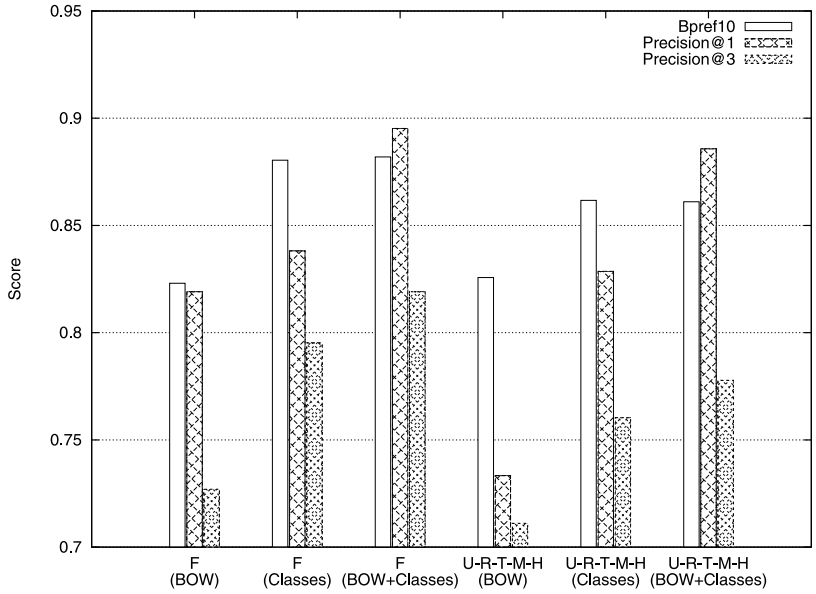

Fig. 15. The effect of classification for Dataset 1. F corresponds to the full page and U-R-T-M-H to the use of the URL, referrer URL, title, metadata, and headings sections of the page. (BOW) shows the effect without classification ($\alpha = 1, \beta = 0$), (Classes) the effect of the classification, that is, only the classification features are used without the corresponding text information ($\alpha = 0, \beta = 1$), and (BOW+Classes) to the effect when both the text and the classification features are used ($\alpha = 1, \beta = 1$).

order of the scores of the various approaches is almost always the same in both cases.

4.9. Web Page Summarization Using MEAD

Here we see the results when we use MEAD for text summarization. In Figures 17 and 18 we see how using summaries compares to the use of full text information, as well as to the use of the Web-page-specific excerpts that we mentioned previously. Notice that summaries with 50 words perform poorly, while taking large summaries (500 words) manages to overtake the simple Web extraction techniques, although it still performs inferiorly compared to using the entire Web page.

A more detailed picture for the performance of MEAD-based summaries is shown in Figures 19 and 20. Not surprisingly, the performance increases as the number of words requested increases, but in generally the performance is comparable to the use of our simple and fast techniques, which indicates the suitability of our approach. Actually our results confirm previous findings in text summarization that specific parts of a document such as the title of the first paragraphs are very good summary candidates.

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. A recent study [Wang et al. 2002] confirms the intuition that ads

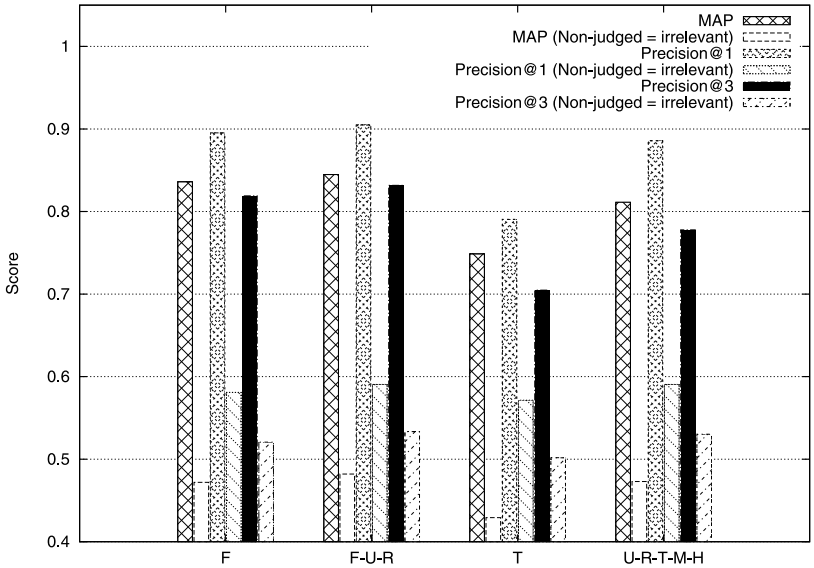

Fig. 16. Comparison of scores when nonjudged ads are ignored and when they are taken into account and considered as irrelevant in Dataset 1. F corresponds to the full page, U corresponds to the page's URL, R to the referrer page's URL, T to the page's title, M to the metadata, and H to the headings sections of the page. The results are with classification. We can see that the two sets of scores are highly correlated, in particular, the relative order of the scores is almost always the same whether we take nonjudged ads into account or not.

need to be relevant to the user's interests in order to avoid degrading the user experience and to increase the probability of reaction.

Ribeiro-Neto et al. [2005] 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 [Lacerda et al. 2006], 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 Ribeiro-Neto et al. [2005].

While these two techniques take advantage of multiple features from the page, they lead to situations where the ad is placed on the page based on the occurrence of one

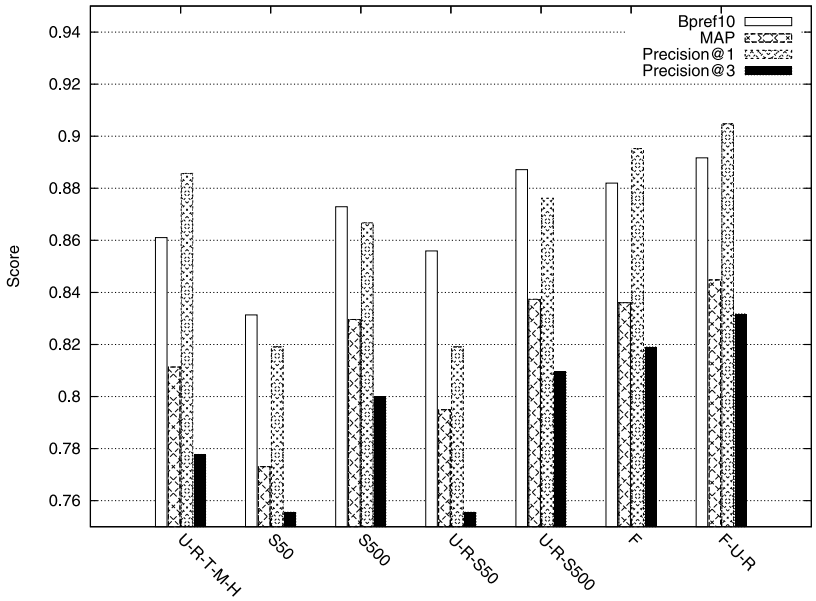

Fig. 17. Comparison of Web-based summarization with MEAD-based summarization for Dataset 1. U corresponds to the page's URL, R to the referrer page's URL, T to the page's title, M to the metadata, H to the headings sections of the page, and F corresponds to the full page. S50 and S100 correspond to the MEAD summarization with summaries consisting of 50 and 500 words, respectively. The results are with classification.

or more ambiguous phrases that might have strong impact on the score. For example, a page about a famous golfer "John Maytag" might trigger an ad for the appliances of the brand with the same name. Another example could be a page describing the Chevy Tahoe SUV triggering a ski trip to Lake Tahoe advertising; words such as "Tahoe" and "Maytag" are proper names and will have relatively high weight compared to the rest of the words in the page and the ads.

In order to solve this problem, Broder et al. [2007b] proposed an ad selection method that combines a semantic phase with the traditional keyword matching (syntactic phase). The semantic phase classifies the page and the ads into a taxonomy of topics and uses the proximity of the ad and page classes as a factor in the ad ranking formula. The result is ads that are topically related to the page. In this manner, one can avoid the pitfalls of the purely syntactic techniques. Furthermore, using a taxonomy one allows for generalization of the search space in the case when there are no ads matching the topic of the page. For example, if the page is about a curling event, a pretty rare winter sport, and contains the words "Alpine Meadows", the system would still rank highly ads for skiing in Alpine Meadows as these are classified in a class "skiing", which is a sibling of the class "curling", and both of these classes share the parent "winter sports".

In this approach, the classes of the page are used to select the set of applicable ads and the keywords are used to further narrow down the search to topics that are too specific to be included in the taxonomy. The taxonomy contains nodes for topics that do not change fast, as for example, brands of digital cameras. The keywords capture the specificity on the level that is more dynamic, and which cannot be captured in the taxonomy due to the huge maintenance overhead. In the digital camera example, this would correspond to the level of particular camera models. Updating the taxonomy each time a new model comes to the market is prohibitively expensive.

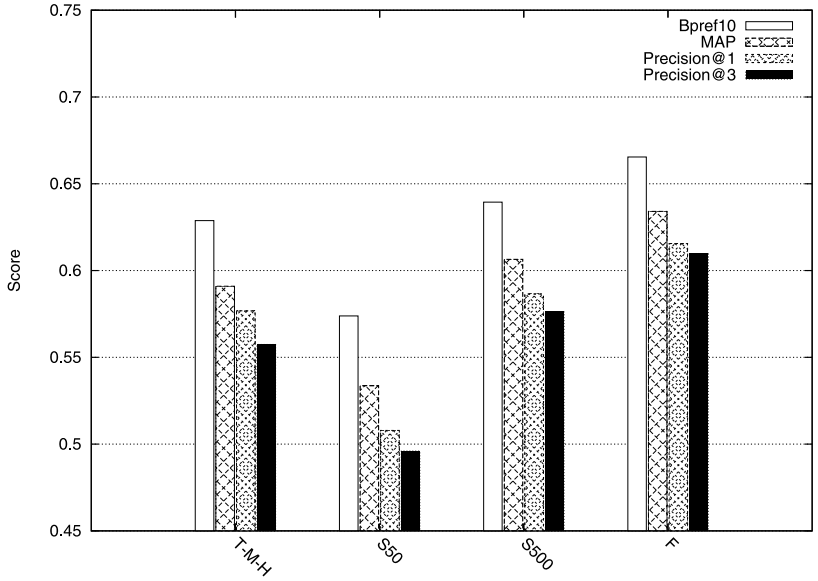

Fig. 18. Comparison of Web-based summarization with MEAD-based summarization for Dataset 2. T corresponds to the page's title, M to the metadata, H to the headings sections of the page, and F corresponds to the full page. S50 and S100 correspond to the MEAD summarization with summaries consisting of 50 and 500 words, respectively. The results are with classification.

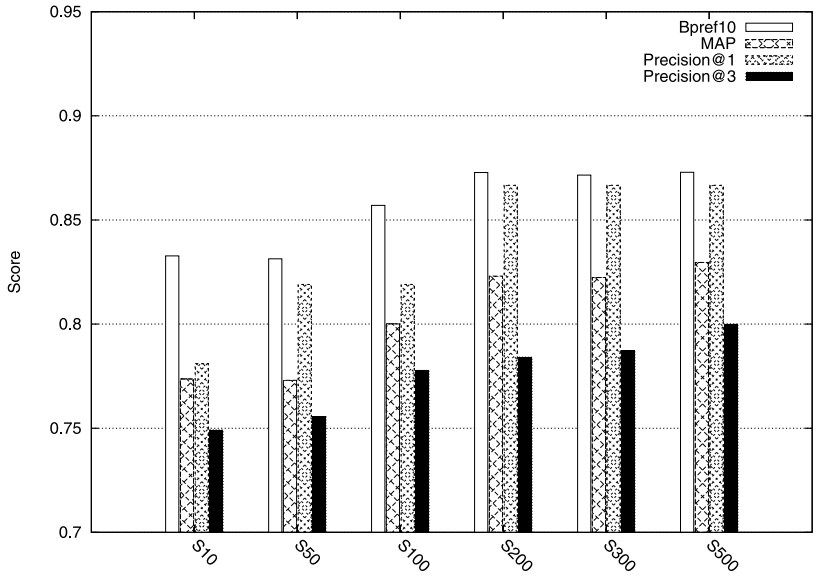

Fig. 19. Performance of the MEAD summarization as the length of the summaries increases for Dataset 1. The values in the x-axis (denoted by Sx) correspond to the number of words requested. The results are with classification.

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. [2006] described a system for phrase

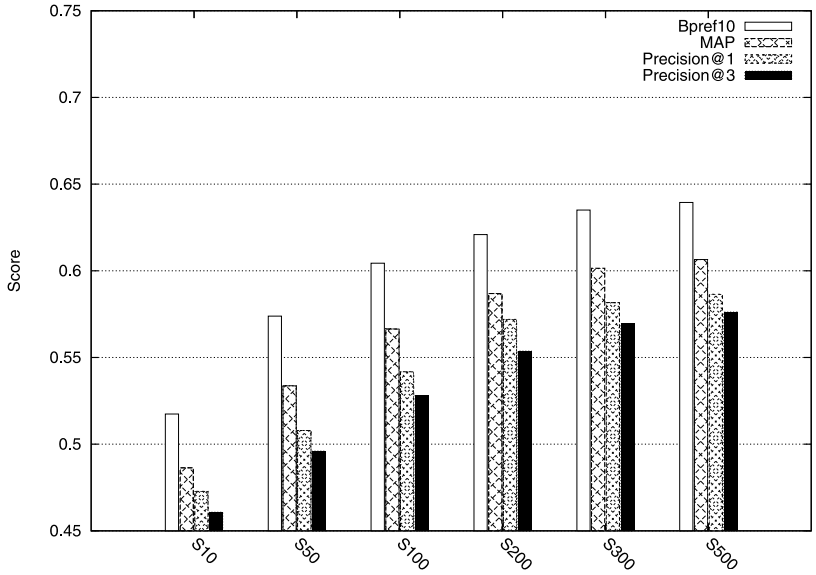

Fig. 20. Performance of the MEAD summarization as the length of the summaries increases for Dataset 2. The values in the x-axis (denoted by Sx) correspond to the number of words requested. The results are with classification.

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 metadata, 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. Broder et al. [2007b] experimented with a phrase extractor developed by Stata et al. [2000]; however, while slightly increasing the precision, it did not change the relative performance of the explored algorithms.

Langheinrich et al. [1999] 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.

Chakrabarti et al. [2008] propose a method for extracting and selecting features from ads and Web pages. They use them to learn scoring models based on logistic regression from historic click-through data.

With the exception of the studies by Yih et al. [2006] and Chakrabarti et al. [2008], all prior works mostly experimented with the different parts of the ad, assuming the publisher's page is given in its entirety. The study of Yih et al. [2006] did take into account the different page parts (e.g., title, metadata, and specific location of the text on the page), but they used them for a completely different task, namely, identifying good advertising keywords. Chakrabarti et al. [2008] studied a different task as well, the design of scoring models to increase relevance. 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.

The method proposed in this article is also related to several sponsored search advertising approaches. Broder et al. [2007a] proposed a query classification method that circumvents the query shortness issue by using the content of the Web pages returned by a search engine for the given query. That work also explored several

summarization approaches for the resulting Web pages, showing that only a small portion of the page can be used without large impact on query classification performance. A related approach has been used to cast the problem of sponsored search as that of content match by placing ads on the Web search result page [Broder et al. 2008]. To overcome the shortness of the query in that approach, the content of the search result pages composed of multiple Web page snippets is used to provide additional features for ad matching. Here we can again see that summarization of the content of the Web pages preserves the ad matching performance while reducing the number of required features. Using Web search results for query expansion has also been successfully applied to query rewriting for sponsored search [Broder et al. 2009; Radlinski et al. 2008].

Finally, the core task in content match approaches that use multiple features for ad retrieval is to perform the similarity search over a large corpus of ads in a high-dimensional space. Unlike most traditional search problems, the ad corpus is defined hierarchically in terms of advertiser accounts, campaigns, and ad groups, which further consist of creatives and bid terms. This hierarchical structure makes indexing highly nontrivial, as naïvely indexing all possible "displayable" ads leads to a prohibitively large and ineffective index. Bendersky et al. [2010] showed that ad retrieval using such an index is not only slow, but its precision is suboptimal as well. They further explored various strategies for compact, hierarchy-aware indexing of textual ads through adaptation of standard IR indexing techniques, and proposed a new ad retrieval method that yields more relevant ads by exploiting the structured nature of the ad corpus. Proposed methods were shown highly effective and efficient compared to the standard indexing and retrieval approaches.

5.2. Classical Text Summarization

Text summarization research has started as early as in the 50's. There are two main approaches, the *abstractive*, where the system creates a summary by extracting information from the document and formulating text, and the *extractive*, where the summarizer simply extracts sentences from the document with high information content. Most research has focused on extractive methods as the quality of summaries that they produce is higher (due to the hard problem of sentence formulation) and because users generally prefer seeing the sentences in the form that the author created. Furthermore, research has focused on *single-document* summarization, and on *multidocument* summarization, where the goal is to extract a summary from an entire collection of documents.

A nice description of various approaches to text summarization appears in the survey by Das and Martins [2007]. In single-document summarization, the first techniques that appeared in the literature [Edmundson 1969; Luhn 1958a, 1958b] scored document sentences based on features such as frequency, *TFIDF*, sentence position, whether the sentence appears in a heading, and so on. Later in the 90's, researchers started applying machine learning techniques. The first models were based on independence of features assumptions and on naive Bayes classification [Aone et al. 1999; Kupiec et al. 1995]. Other attempts used more complicated models such as decision trees [Lin 2000], hidden Markov models [Conroy and O'Leary 2001], log-linear models [Osborne 2002], neural networks [Svore et al. 2007], and matrix factorization techniques [Lee et al. 2009]. Another line of research attempts to model the document's structure using natural language analysis techniques [Barzilay and Elhadad 1997; Marcu 1998; Ono et al. 1994]. Finally, Li et al. [2009] create document summaries by explicitely trying to maximize the diversity and the coverage of the document achieved by the summary created. Multidocument summarization gained

popularity due to the use in the domain of news articles. One of the main ideas on which several techniques were based is the construction of a graph where nodes correspond to words/sentences/concepts and edges correspond to distance between concepts, and the goal is to select nodes that are close to the center of the graph or nodes that are dispersed in the graph [Antiqueira et al. 2009; Erkan and Radev 2004; Mani and Bloedorn 1997].

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. [2002], and later Alam et al. [2003] and Otterbacher et al. [2008] studied summarization of Web pages for presentation on handheld devices. Sun et al. [2005] summarized Web pages by using click-through 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 [Jatowt and Ishizuka 2004]. 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 [2000] 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 nonextractive summarization by "translating" a page using techniques based on statistical machine translation.

More recently, summarization methods have started taking into account the features of the so-called social Web or Web 2.0. This means that they take into account users' explicit feedback through the use of commenting and tagging systems. Park et al. [2008] developed a basic system for summarizing Web documents using user comments and annotations from delicious. Hu et al. [2008] also used user comments to assist summarization and they experimented with various methods for weighting the comments and creating the summary. Zhu et al. [2009] used information from user tags to assist summarization, and to that purpose they developed a HITS-type of ranking method. Finally, Boydell and Smyth [2010] use tag information from delicious to assist to the extraction of snippets from Web documents.

Several works studied the synergy between text summarization and text classification. Kolcz et al. [2001] used summaries to perform feature selection, assuming that terms that occur in the summary are more informative for categorization. Shen et al. [2004] 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 metafields (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 parts 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 with minimal processing overhead. 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 sources 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 yields 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 6% 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. Finally, we compared our Web-page-specific page extraction approach with more advanced text summarization techniques and we observed that the two methods have comparable performance for the problem of content matching advertising, while the summaries produced with the former are much smaller.

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. Some page fragments, such as metadata and referrer page's URL, are not always available so their contribution might not be significant overall, however, when available they can significantly increase performance and thus should be included. Referrer page's URL often contains a user query thus it can provide valuable information. For different types of pages, different parts of the page might be more valuable. For example, 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. Thus, a future direction for this work would be being able to classify online the type of page and construct the appropriate summary.

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 and Donald Metzler for useful discussions.

REFERENCES

ALAM, H., HARTONO, R., KUMAR, A., RAHMAN, F., TARNIKOVA, Y., AND WILCOX, C. 2003. Web page summarization for handheld devices: A natural language approach. In *Proceedings of the International Conference on Document Analysis and Cognition (ICDAR'03)*.

ANAGNOSTOPOULOS, A., BRODER, A., GABRILOVICH, E., JOSIFOVSKI, V., AND RIEDEL, L. 2007. Just-in-time contextual advertising. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*. 331–340.

ANTIQUEIRA, L., JR., O. N. O., DA FONTOURA COSTA, L., AND DAS GRAAS VOLPE NUNES, M. 2009. A complex network approach to text summarization. *Inf. Sci. 179*, 5, 584–599.

AONE, C., GORLINSKI, J., LARSEN, B., AND OKUROWKSI, M. E. 1999. A trainable summarizer with knowlege acquired from robust NLP techniques. In *Advances in Automatic Text Summarization*.

BARZILAY, R. AND ELHADAD, M. 1997. Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop*.

BENDERSKY, M., GABRILOVICH, E., JOSIFOVSKI, V., AND METZLER, D. 2010. The anatomy of an ad: Structured indexing and retrieval for sponsored search. In *Proceedings of the 19th International World Wide Web Conference*.

BERGER, A. AND MITTAL, V. O. 2000. OCELOT: A system for summarizing Web pages. In *Proceedings of the SIGIR'00 Conference*.

BOYDELL, O. AND SMYTH, B. 2010. Social summarization in collaborative web search. *Inf. Process. Manage. 46*, 782–798.

BRIN, S. AND PAGE, L. 1998. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the International Conference on World Wide Web (WWW'98)*.

BRODER, A., FONTOURA, M., GABRILOVICH, E., JOSHI, A., JOSIFOVSKI, V., AND ZHANG, T. 2007a. Robust classification of rare queries using web knowledge. In *Proceedings of the SIGIR'07 Conference*. 231–238.

BRODER, A., FONTOURA, M., JOSIFOVSKI, V., AND RIEDEL, L. 2007b. A semantic approach to contextual advertising. In *Proceedings of the SIGIR'07 Conference*. ACM Press, 559–566.

BRODER, A., CICCOLO, P., FONTOURA, M., GABRILOVICH, E., JOSIFOVSKI, V., AND RIEDEL, L. 2008. Search advertising using Web relevance feedback. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM'08)*. 1013–1022.

BRODER, A., CICCOLO, P., GABRILOVICH, E., JOSIFOVSKI, V., METZLER, D., RIEDEL, L., AND YUAN, J. 2009. Online expansion of rare queries for sponsored search. In *Proceedings of the 18th International World Wide Web Conference*. 511–520.

BUCKLEY, C. AND VOORHEES, E. M. 2004. Retrieval evaluation with incomplete information. In *Proceedings of the SIGIR'04 Conference*.

BUYUKKOKTEN, O., KALJUVEE, O., GARCIA-MOLINA, H., PAEPCKE, A., AND WINOGRAD, T. 2002. Efficient web browsing on handheld devices using page and form summarization. *ACM Trans. Inf. Syst. 20*, 1, 82–115.

CHAKRABARTI, D., AGARWAL, D., AND JOSIFOVSKI, V. 2008. Contextual advertising by combining relevance with click feedback. In *Proceedings of the International Conference on World Wide Web (WWW'08)*. 417–426.

CHATTERJEE, P., HOFFMAN, D. L., AND NOVAK, T. P. 2003. Modeling the clickstream: Implications for web-based advertising efforts. *Market. Sci. 22*, 4, 520–541.

CONROY, J. AND O'LEARY, D. P. 2001. Text summarization via hidden markov models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*. W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel Eds. ACM Press, New York, 406–407.

DAS, D. AND MARTINS, A. 2007. A survey on automatic text summarization. http://www.cs.cmu.edu/~afm/Home.html.

DUDA, R. AND HART, P. 1973. *Pattern Classification and Scene Analysis*. John Wiley and Sons.

EDELMAN, B., OSTROVSKY, M., AND SCHWARZ, M. 2007. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *Amer. Econ. Rev. 97*, 1, 242–259.

EDMUNDSON, H. P. 1969. New methods in automatic extracting. *J. ACM 16*, 2, 264–285.

ERKAN, G. AND RADEV, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res. 22*, 457–479.

FAIN, D. AND PEDERSEN, J. 2006. Sponsored search: A brief history. In *Proceedings of the 2nd Workshop on Sponsored Search Auctions*.

GABRILOVICH, E. AND MARKOVITCH, S. 2005. Feature generation for text categorization using world knowledge. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*. 1048–1053.

GATES, S. C., TEIKEN, W., AND CHENG, K.-S. F. 2005. Taxonomies by the numbers: Building high-performance taxonomies. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM'05)*. ACM Press, New York, 568–577.

HAN, E.-H. S. AND KARYPIS, G. 2000. Centroid-Based document classification: Analysis and experimental results. In *Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases*.

HAWKING, D., CRASWELL, N., AND THISTLEWAITE, P. 1998. Overview of TREC-7 very large collection track. In *Proceedings of the TREC-7 Conference*.

HU, M., SUN, A., AND LIM, E.-P. 2008. Comments-oriented document summarization: Understanding documents with readers' feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, 291–298.

JATOWT, A. AND ISHIZUKA, M. 2004. Web page summarization using dynamic content. In *Proceedings of the International Conference on World Wide Web (WWW'04)*.

KOLCZ, A., PRABAKARMUTHI, V., AND KALITA, J. 2001. Summarization as feature selection for text categorization. In *Proceedings of the SIGIR'01 Conference*. 365–370.

KUPIEC, J., PEDERSEN, J., AND CHEN, F. 1995. A trainable document summarizer. In *Proceedings of the 18th Annual International Conference on Research and Development in Information Retrieval (SIGIR'95)*, E. A. Fox, P. Ingwersen, and R. Fidel Eds. ACM Press, New York, 68–73.

LACERDA, A., CRISTO, M., GONCALVES, M. A., FAN, W., ZIVIANI, N., AND RIBEIRO-NETO, B. 2006. Learning to advertise. In *Proceedings of the SIGIR'06 Conference*. 549–556.

LANGHEINRICH, M., NAKAMURA, A., ABE, N., KAMBA, T., AND KOSEKI, Y. 1999. Unintrusive customization techniques for web advertising. *Comput. Netw. 31*, 1259–1272.

LEE, J.-H., PARK, S., AHN, C.-M., AND KIM, D. 2009. Automatic generic document summarization based on non-negative matrix factorization. *Inf. Process. Manage. 45*, 20–34.

LI, L., ZHOU, K., XUE, G.-R., ZHA, H., AND YU, Y. 2009. Enhancing diversity, coverage and balance for summarization through structure learning. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09)*. ACM, New York, 71–80.

LIN, C.-Y. 2000. Training a selection function for extraction. In *Proceedings of the 8th International Conference on Information Knowledgement (CIKM'99)*. ACM Press, 55–62.

LUHN, H. P. 1958a. The automatic creation of literature abstracts. *IBM J. Res. Devel. 2*, 159–165.

LUHN, H. P. 1958b. The automatic creation of literature abstracts. *IBM J. Res. Devel. 2*, 159–165.

MANI, I. AND BLOEDORN, E. 1997. Multi-Document summarization by graph search and matching. In *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI-97/IAAI-97)*. AAAI Press, 622–628.

MARCU, D. 1998. Improving summarization through rhetorical parsing tuning. In *Proceedings of the 6th Workshop on Very Large Corpora*. 206–215.

ONO, K., SUMITA, K., AND MIIKE, S. 1994. Abstract generation based on ahetorical structure extraction. In *Proceedings of the 15th conference on Computational Linguistics*. Association for Computational Linguistics, 344–348.

OSBORNE, M. 2002. Using maximum entropy for sentence extraction. In *Proceedings of the ACL'02 Workshop on Automatic Summarization*. Association for Computational Linguistics, 1–8.

OTTERBACHER, J., RADEV, D., AND KAREEM, O. 2008. Hierarchical summarization for delivering information to mobile devices. *Inf. Process. Manag. 44*, 2, 931–947.

PARK, J., FUKUHARA, T., OHMUKAI, I., TAKEDA, H., AND LEE, S.-G. 2008. Web content summarization using social bookmarks: A new approach for social summarization. In *Proceeding of the 10th ACM Workshop on Web Information and Data Management (WIDM'08)*. ACM, New York, 103–110.

POWERS, S. 2008. *Learning JavaScript* 2nd Ed. O'Reilly Media.

RADEV, D. R., TEUFEL, S., SAGGION, H., LAM, W., BLITZER, J., QI, H., CELEBI, A., LIU, D., AND DRABEK, E. 2003. Evaluation challenges in large-scale multi-document summarization: The MEAD project. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*.

RADLINSKI, F., BRODER, A., CICCOLO, P., GABRILOVICH, E., JOSIFOVSKI, V., AND RIEDEL, L. 2008. Optimizing relevance and revenue in ad search: A query substitution approach. In *Proceedings of the SIGIR'08 Conference*. 403–410.

RIBEIRO-NETO, B., CRISTO, M., GOLGHER, P. B., AND DE MOURA, E. S. 2005. Impedance coupling in content-targeted advertising. In *Proceedings of the SIGIR'05 Conference*.

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

SALTON, G. AND BUCKLEY, C. 1988. Term weighting approaches in automatic text retrieval. *Inf. Process. Manag. 24*, 5, 513–523.

SHEN, D., CHEN, Z., ZENG, H.-J., ZHANG, B., YANG, Q., MA, W.-Y., AND LU, Y. 2004. Web-page classification through summarization. In *Proceedings of the SIGIR'04 Conference*.

STATA, R., BHARAT, K., AND MAGHOUL, F. 2000. The term vector database: fast access to indexing terms for Web pages. *Comput. Netw. 33*, 1–6, 247–255.

SUN, J.-T., SHEN, D., ZENG, H.-J., YANG, Q., AND LU, YUCHANG ANF CHEN, Z. 2005. Web-page summarization using clickthrough data. In *Proceedings of the SIGIR'05 Conference*. 194–201.

SVORE, K., VANDERWENDE, L., AND BURGES, C. 2007. Enhancing single-document summarization by combining RankNet and third-party sources. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07)*. 448–457.

W3C. 2005. Document object model, level 1 specification. http://www.w3.org/TR/REC-DOM-Level-1/.

WANG, C., ZHANG, P., CHOI, R., AND EREDITA, M. D. 2002. Understanding consumers attitude toward advertising. In *Proceedings of the 8th Americas Conference on Information Systems*.

YIH, W.-T., GOODMAN, J., AND CARVALHO, V. R. 2006. Finding advertising keywords on Web pages. In *Proceedings of the International Conference on World Wide Web (WWW'06)*.

ZHU, J., WANG, C., HE, X., BU, J., CHEN, C., SHANG, S., QU, M., AND LU, G. 2009. Tag-oriented document summarization. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, New York, 1195–1196.

Received September 2010; revised December 2010; accepted February 2011