Essential Pages

Ashwin Swaminathan[†], Cherian Varkey Mathew[‡], and Darko Kirovski[◇]
[†] University of Maryland, College Park, MD, USA
[‡] Indian Institute of Technology, Kanpur, India
[◇] Microsoft Research, Redmond, WA, USA

Contact: darkok@microsoft.com

Technical Report MSR-TR-2008-015
January 2008

Microsoft Research
One Microsoft Way Redmond, WA 98052, USA
http://research.microsoft.com

ESSENTIAL PAGES

ABSTRACT

Results to Web search queries are ranked using heuristics that typically analyze the global link topology, user behavior, and content relevance. We point to a particular inefficiency of such methods: information redundancy. In queries where learning about a subject is an objective, modern search engines return relatively unsatisfactory results as they consider the query coverage by each page individually, not a set of pages as a whole.

We address this problem using essential pages. If we denote as $\mathbb{S}_Q$ the total knowledge that exists on the Web about a given query Q, we want to build a search engine that returns a set of essential pages E_Q that maximizes the information covered over $\mathbb{S}_Q$. In this paper, we present a preliminary prototype that optimizes the selection of essential pages; we draw some informal comparisons with respect to existing search engines; and finally, we demonstrate our prototype in action using a blind-test user study.

1. INTRODUCTION

Web-search, although itself not a money-maker, is certainly one of the premium applications on the Internet, resulting in substantial ad revenues. Results to Web-search queries are typically influenced by several metrics [1]:

{C} **content relevance** [2] derived from documents' anchor text [1], title and headings, word frequency and proximity [3], file, directory, and domain names, and other more sophisticated forms of content analysis.

{U} **user behavior** extrapolated from user's time-spent-on-page, time-on-domain, clickthrough rates, etc. [4].

{P} **popularity** in the global link structure [5, 6, 7] with authority [8], readability [9], and novelty [10] typically determining the linkage.

Links to the most "relevant" pages, according to the above criteria, are then potentially clustered [11] and delivered to users who in turn browse the results to find the desired information. Although researched in detail along most of the mentioned criteria, search engines still leave a lot to be desired. In this paper, we emphasize one important inefficiency of state-of-the-art search engines: content redundancy, and propose a system that significantly improves search results for learning-type queries. Looking from a user's perspective, we review an existing classification of Web-search queries [12, 13] that aim at predominantly textual content (i.e., non-multimedia):

- **navigational** – the user is seeking a Web-site with an unknown URL; typically, a single specific Web-site is the "correct answer" to the posed query (e.g., `washington mutual` points to `www.wamu.com`).
- **journey** – the user is browsing a certain content category on the Web at random trying to discover further points of interest (e.g., `indoor plants`).
- **shopping** – the user is looking for the best offer on the Web for a product/service (e.g., `nikon d300 price`).
- **learning** – the user wants to know a specific detail or the entire breadth of knowledge available on the Web for a specific query (e.g., `rhododendron`).

While queries from the first three categories could be handled with relatively simple URL lists: a single "correct" URL, a randomized list of "relevant" pages, and an unbiased comparative shopper, learning-type queries remain difficult to address. The main culprit is the cumulative information redundancy over the ranked list of returned documents. It dominates search results to the point where finding content that is not displayed on an encyclopedia-style Web-page such as Wikipedia, usually ranked at the very top of results, is a cumbersome task that requires browsing dozens of links. In this paper, our goal is to propose a simple yet powerful tool for pass-filtering a small set of text documents that offers the greatest **joint** coverage on a given topic.

If we denote as $\mathbb{S}_Q$ the total knowledge that exists on the Web about a given query Q, we want to build a search engine that returns a set of essential pages[1] E_Q that maximizes the information covered over $\mathbb{S}_Q$. While $\mathbb{S}_Q$ is truly a semantic digest of all Web content related to Q, we argue that a simple "bag-of-words" approach to representing $\mathbb{S}_Q$ is a surprisingly efficient model. Then, we formalize the overall optimization objective using a weighted coverage function that takes into account both word and page relevance. Using the Sequential Forward Floating Selection (SFFS) algorithm [14], we show that a fast URL ordering by their **joint** knowledge coverage is achievable and well accepted by users.

Figure 1 illustrates an abstract example of how $\mathbb{S}_Q$ is covered by a set of pages computed using a traditional page ranking algorithm (top) and a set of essential pages assembled to maximize their joint query coverage (bottom). As

[1]We have borrowed the expression "essential" from terminology used in logic synthesis where a binary minterm not fully covered by the remaining minterms of a containing binary function is called an essential minterm.

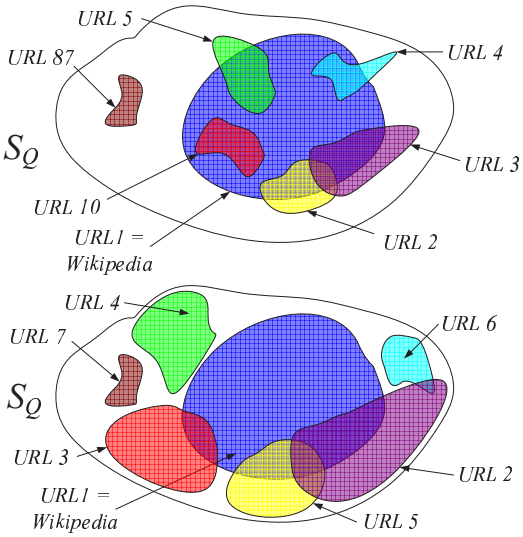

Figure 1: An abstract example of query coverage by an existing (top) and the proposed page ranking algorithm (bottom).

a result, in the traditional model, in order to learn details about $\mathbb{S}_Q$ users have to browse substantially more pages.

In a simple implementation, we used an existing search engine to obtain a large list of most relevant URLs for a given query, then used our tool for post-processing, i.e., reordering, of these results. This way, we avoided building and running a Web crawler and a Web index. In an overwhelming majority of cases, our ranking was substantially different than the ranking returned by the underlying search engine. In a user study with over 120 search queries and 35 subjects, in approximately 5 out of 6 learning queries users found our ranking to provide better or equal learning experience to Google's. This result is impressive from the perspective that we did not use[2] any other metric from the **{C,U,P}** set to improve our rankings. Our optimization strategy is orthogonal to the **{C,U,P}** methods in its objective and can be combined with them in an arbitrary manner to achieve superior page ranking results.

1.1 Related Work

The volume of research conducted on the topic of Web-search is enormous. Thus, in this section we review a minimal, most related scope of work, otherwise not mentioned throughout our paper. Recently, Chen and Karger presented a method to rank documents using an optimization framework to maximize the probability of finding a relevant document in the top n [15]. They have shown that the global optimization problem is NP-hard and proposed an efficient heuristic to address it. To the best of our knowledge, this is the only effort to date that addresses in an optimization-intensive manner the quality of results (i.e., Web pages) to Web-search queries as a whole, not individually. In the case of Chen and Karger's work, they addressed result relevance as the sole optimization objective.

[2]Except for obtaining the initial large list of candidate URLs from an existing search engine. The fact that this list was relatively large attenuates the effect of its ranking on our technology.

1.1.1 Result Diversity

Existing literature has also considered diversity of Web-search results as an additional factor for ordering documents. A re-ranking technique was proposed in [16] based on the maximum marginal relevance criterion to reduce redundancy from search results as well as to present document summarizations. Zhai et al. have defined the subtopic retrieval problem as finding documents that cover as many different subtopics of a general topic as possible [17]. In [18], the authors propose an affinity ranking scheme to re-rank search results by optimizing diversity and information richness of the topic and query results.

The aforementioned three techniques [16, 17, 18] model the variance of topics in groups of documents. They all have difficulties applying the concept of diversity to Web-search, e.g., in [18], the authors assume that all pages are labeled with the topics they cover, then rank them to roughly improve the number of topics covered by a set of pages. Clearly, in the most applicable case of Web-search, such labeling is not available a priori. For that reason, we do not compare our search engine to [16, 17, 18] experimentally, rather we chose to compare our results to state-of-the-art search engines such as Google's, for which we speculate that they address the problem of result diversity.

In contrast to prior known methods that focus on maximizing diversity, the technology introduced in this work aims at modeling the overall finite knowledge space for a specific query and improving the coverage of this space by a set of documents. We propose a "bag-of-words" model for representing knowledge spaces, introduce a formal notion of coverage over the "bag-of-words," and derive a simple but systematic algorithm to select documents that maximize coverage, while being relevant to the search topic. As a side-effect, our approach results in improved "diversity" [16] (i.e., "subtopic retrieval" [17] or "information richness" [18]) of the resulting document ranking.

1.1.2 Clustering

One could consider clustering of Web-search results as related to the objective of essential pages [11]. We argue that the objectives are orthogonal. A possible strategy to enforce diversity is to cluster the results and return the most relevant Web-page(s) from each cluster. Such an approach is overly ad-hoc, does not consider the knowledge related to a specific query as a finite set of information, and assumes that content in different Web-pages does not partially overlap. Essential pages not only formally define and address the problem of knowledge coverage, but could simplify clustering substantially. Namely, using essential pages as preprocessing to clustering should greatly reduce the workload for the clustering algorithm which should now look for non-overlapping subsets of essential pages as cluster nodes, as opposed to analyze a large number of partially redundant documents relevant to the query [11].

2. RELEVANCE-BASED SEARCH

We first review a traditional relevance-based search mechanism related to our technology. Consider a database $\mathbb{D}$ of N_d documents. The goal of relevance-based rank-ordered search is to generate a permutation $\pi_Q(\mathbb{D})$ based on a search query Q so that the documents that have higher relevance to Q, come higher in the retrieval results.

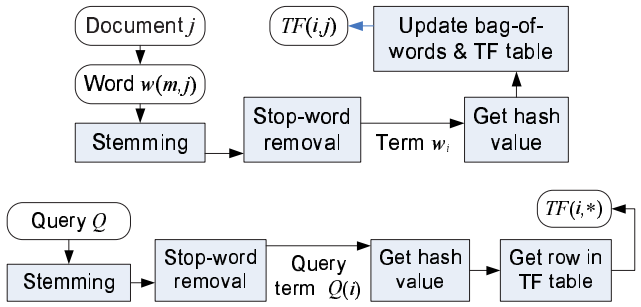

Figure 2: Document pre-processing and indexing to generate the term-frequency table (top). Query processing steps in relevance-based search engines (bottom).

Figure 2 shows the rank-ordered search process present in many search engines [19, 20]. For each document j in $\mathbb{D}$ ($1 \leq j \leq N_d$), all words in the document are first extracted. The m-th word in the j-th document, $w(m, j)$, then undergoes stemming [21]. In this step, the word root is retained while word endings are removed. Words such as `as`, `is`, `be`, etc., in a pre-defined set of stop-words are then removed as they do not describe the context semantics [22, 23]. Stemming and stopping improve search performance by giving users more pertinent results; they also reduce the search complexity by reducing the dictionary of words.

We denote the total number of unique terms in the resulting list $\mathbb{T}$ as N_t. Term frequency $TF(i, j)$ indicates the number of times the i-th term appears in the j-th document. The term frequency information for $\mathbb{D}$ and $\mathbb{T}$ is organized as a term frequency table of size $N_t \times N_d$. To facilitate fast access, a hash table is constructed to map each term to the corresponding row of the term frequency table.

The term frequency data provides valuable information about document relevance, and is employed as a core variable to define the relevance score in rank-ordering documents. One example metric is the Okapi relevance score $CW(i, j)$ [19], which is defined as:

$$CW(i,j) = \frac{(K_1+1)\, CFW(i)\, TF(i,j)}{K_1\,[1-b+b\cdot NDL(j)] + TF(i,j)}. \quad (1)$$

Here, $CFW(i)$ denotes the cumulative frequency of the i-th term in the entire database, and is obtained as:

$$CFW(i) = -\log_2(n_i/N_d),$$

where n_i is the total number of documents in $\mathbb{D}$ which contains the i-th term. $NDL(j)$ in Eq. 1 represents the normalized length of the j-th document and is computed by dividing the length of the j-th document, $L(j)$, by the average document length L_{avg}, i.e., $NDL(j) = L(j)/L_{avg}$. Constants K_1 and b could be tailored to meet the needs of specific types of databases; some example values are $K_1 = 2$ and $b = 0.75$. The term frequency table and an adopted relevance score metric are the two main tools used in relevance-based search engines.

A single-word search query Q is handled as follows. First, the query word Q undergoes stemming and stopping. Then, its hash value is used to point to the corresponding row of the term frequency table. Documents in $\mathbb{D}$ are then sorted in decreasing order of their relevance score computed using Eq. 1. This process is shown in Figure 2. If a query Q contains multiple terms $\{Q(1), \ldots, Q(N_q)\}$, we compute the relevance score $\mathcal{R}(j)$ for document j as follows:

$$\mathcal{R}(j) = \sum_{i=1}^{N_q} CW(k_i, j). \quad (2)$$

The relevance score $\mathcal{R}(j)$ is computed for all documents in $\mathbb{D}$ and used to rank-order $\mathbb{D}$ in response to a multi-term query.

3. SELECTING ESSENTIAL PAGES

In this section, we propose a novel search engine that aims to find a set of pages that gives maximum coverage about a particular search query Q over the related knowledge space $\mathbb{S}_Q$. Ideally, finding a set of essential pages E_Q that maximally covers $\mathbb{S}_Q$ requires a careful semantic analysis of each Web-page, a task which demands cumbersome and careful human intervention. We argue that a simple "bag-of-words" approach to this problem, already deployed in relevance-based rank-ordered search engines, performs surprisingly well. In subsequent subsections, we introduce the approach and define a coverage score metric that we found to perform particularly well in our experiments.

3.1 The Bag-of-Words Approach

The bag-of-words approach is a technique widely deployed in information retrieval literature [20, 22]. The idea is to treat a document as a collection of statistics over the set, i.e., bag, of words used in it, without any explicit semantic constructs such as sentences, formatting, etc. We rely on this paradigm to describe the knowledge base $\mathbb{S}_Q$ related to a query Q. Intuitively, unless semantically important distinct information in $\mathbb{S}_Q$ is described using permutations over the same set of words which is a rarity, "bag-of-words" should provide a solid building block for our application. Just as in relevance-based document ordering, we consider each Web-page (i.e, document) as a bag-of-words: each distinct word is associated with the total number of times it appears in a specific document.

Document indexing for essential pages is equivalent to the process illustrated in Figure 2. For each page on the Web, its distinct set of words is extracted, stemmed using [21], and filtered for stop words [23]. The global term-frequency table is computed and stored as presented in Section 2. Given a single-term query Q, the subset of documents, $\mathbb{D}_Q$, containing Q is identified using the global term-frequency table. As $\mathbb{D}_Q$ contains all the information about Q, we denote the set of terms (bag-of-words) extracted from $\mathbb{D}_Q$ as $\mathbb{S}_Q$.

We informally formulate the problem of essential page selection as finding a subset of documents $E_Q \subset \mathbb{D}_Q$ that provides maximum coverage about the query, where the formal notion of information coverage is introduced later. Let $N_d^Q = ||\mathbb{D}_Q||$ and $N_t^Q = ||\mathbb{S}_Q||$. We remark that for a single-term query, all documents in $\mathbb{D}_Q$ contain the query term; for queries containing multiple terms $\{Q(1), \ldots, Q(m)\}$, at least one of these terms appears in each document in $\mathbb{D}_Q$. We denote the subset of the global term-frequency table that relates to the search query Q as $TF^Q \equiv \mathbb{S}_Q \times \mathbb{D}_Q$ and record its size as $N_t^Q \times N_d^Q$. For each term, $t \in \mathbb{S}_Q$, relevant to the query, we define a **term-relevance** score, $r(t)$:

$$r(t) = \frac{n_t^Q}{N_d^Q}, \quad (3)$$

where n_t^Q represents the number of documents in $\mathbb{D}_Q$ which contain t. The term-relevance score heuristically measures how relevant t is to Q; the higher the score, the higher the relevance. We use term-relevance as a relevance metric with important notes: it is different from more complex, widely used relevance metrics such as Okapi (see Eq. 1), it is used due to its simplicity and demonstrated semantic effectiveness while used to identify essential pages.

3.2 Coverage Score

We define a coverage score, $\mathcal{C}(j)$, of a document $j \in \mathbb{D}_Q$:

$$\mathcal{C}(j) = \sum_{\forall t_i \in \mathbb{S}_Q} \gamma(t_i) \times TF^Q(i,j). \quad (4)$$

where $TF^Q(i,j)$ represents the term-frequency value of the i-th term, t_i, in document j; $\gamma(t_i)$ in Eq. 4 quantifies the overall importance given to covering t_i in E_Q, henceforth we refer to this metric as the **term-importance score**, and define it as follows:

$$\gamma(t_i) = r(t_i) \log_2 \left[\frac{1}{r(t_i)} \right]. \quad (5)$$

Figure 3 shows the variation of $\gamma(t_i)$ vs. $r(t_i)$. The rationale behind choosing such a metric to describe word-importance is as follows:

- **Low** $r(t_i)$ – words that are less relevant to the query do not provide significant information about the query, and therefore they are less important.
- **High** $r(t_i)$ – words that are very relevant to the query (such as the query words itself) provide more information about the query. However, they appear in most documents containing the query word. Hence, it is of less importance to cover these words among the bag-of-words, and therefore they are assigned a low word-importance score. This explains the reasoning behind $\gamma(t_i) \to 0$ as $r(t_i) \to 1$.
- **Important words** – the remaining words are deemed relatively important; our algorithm aims at covering as many as possible of these words with a fixed-cardinality subset of pages from $\mathbb{D}_Q$. Intuitively, this is the information that occurs relatively often in $\mathbb{D}_Q$ and thus is likely to be semantically related to Q. It also occurs relatively infrequently so that it could be spread over a number of pages and thus it can be inconvenient for a user to search for it using a traditional search engine.

Generalizing on the definition of the coverage score, $\mathcal{C}(j)$, as given in Eq. 4, we define the joint coverage score of a set of documents. Let there be two documents a and b, with the corresponding bags-of-words $\mathbb{S}_Q^a$ and $\mathbb{S}_Q^b$ respectively. The **joint coverage score** $\mathcal{C}(a \cup b)$ of the combined set of documents is given by:

$$\begin{aligned} \mathcal{C}(a \cup b) &= \sum_{\forall t_i \in \mathbb{S}_Q^a \cup \mathbb{S}_Q^b} \gamma(t_i) \cdot \mathcal{T}(i), \\ \mathcal{T}(i) &= \max\{TF^Q(i,a), TF^Q(i,b)\}. \end{aligned} \quad (6)$$

The definition of the joint coverage score for two documents can be generalized to a set of documents of arbitrary cardinality by replacing $\mathcal{T}(i)$ to be the maximum term-frequency value over all documents in the set.

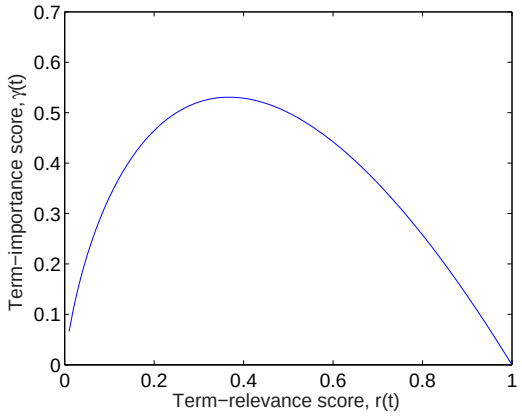

Figure 3: Term-importance score as a function of the term-relevance score.

Note that the coverage metric in Eq. 6 is not normalized by the document length, so it gives an advantage to longer documents. Although for generic search mechanisms such strategy could potentially adversely affect user adoption, for learning-type queries this decision produced results that were well received in the user study.

3.3 The Selection Algorithm

Conventional search engines evaluate the relevance of **individual** Web-pages and present results by sorting them according to a set of specific criteria that excludes mutual dependency [1]. In this paper, we propose that the retrieved results are generated based upon their **joint** coverage of the search query. In addition, we establish an optimization goal for our search engine so that the retrieved result, i.e., list of Web-page pointers, has maximum joint coverage over the query topic. Here we use the definition of joint coverage (see Eqs. 4 and 6) to identify the resulting essential pages.

PROBLEM 1. ***Essential Pages Selection.*** *The objective of the proposed search engine is to identify a set of essential pages, E_Q, of fixed cardinality n_Q that pertains to the query Q as the set of pages that maximizes their joint coverage score. More specifically:*

$$E_Q = \underset{E|E \subset \mathbb{D}_Q, ||E|| = n_Q}{\arg\max} \mathcal{C}\left[\bigcup_{\forall d \in E} d \right]. \quad (7)$$

For a given n_Q, identifying E_Q would require a search of complexity:

$$\mathcal{O}\left[\binom{N_d^Q}{n_Q} \right] \quad (8)$$

under the assumption that the context covered by pages in $\mathbb{D}_Q$ is mutually independent and random. A more complex analysis of problem's complexity[3] is futile because in our application we would like to provide an algorithm which is simple and fast so that retrieval times are not affected.

To address this demand, we have developed a greedy iterative heuristic using the SFFS algorithm [14]. The details of the algorithm are given in Figure 4. We start with $E = \emptyset$ in

[3]In a specific setting, one can show that essential pages selection is a variant of the set covering problem which is NP-hard.

Essential Pages Selection	
1	$E = \emptyset$.
2	**while** $\|\|E\|\| < n_Q$
3	$k = \arg\max_{j \in \mathbb{D}_Q} \mathcal{C}(E \cup j)$
4	**if** $\mathcal{C}(E \cup k) > \mathcal{C}(E)$
5	$E = E \cup k$
6	$m = \arg\max_{i \in E} \mathcal{C}(E - i)$
7	**if** $\mathcal{C}(E - m) \geq \mathcal{C}(E)$
8	$E = E - m$
9	**else continue**
10	**else break**

Figure 4: Pseudo-code for the proposed SFFS-based essential page selection algorithm.

step 1. In each iteration we add one and delete one element from the set to improve coverage. In the adding step shown in step 3, we identify a document $k \subset \mathbb{D}_Q$ which when added to the set E results in maximum coverage. Document k is then added to E in step 5 if $\mathcal{C}(E \cup k) > \mathcal{C}(E)$. In the deletion step 6, a document m which adds the least amount of information to the knowledge space covered by E, is removed from E in step 8 conditional on $\mathcal{C}(E-m) > \mathcal{C}(E)$. This iterative process is repeated until no further improvement can be attained or $||E|| = n_Q$. The computational complexity of this algorithm is $\mathcal{O}(n_Q^2 \cdot ||\mathbb{D}_Q||)$.

4. SYSTEM IMPLEMENTATION

We implemented the proposed system with emphasis on showcasing the new search paradigm and with no other particular engineering objective such as optimizing the speed of query handling. The pre-processing and the indexing phases of the proposed system are as shown in Figure 2. In this stage, for an input query Q we use an existing search engine such as Google or Microsoft's Live Search to retrieve a large set of documents, $\mathbb{D}_Q$, related to Q onto a local machine. In our experiments, we used $N_d^Q = 200$. Then, for each document in $\mathbb{D}_Q$ we remove the stop words and perform stemming using [21]. As output, we build the related bag-of-words $\mathbb{S}_Q$ and the term-frequency matrix TF^Q.

The search stage follows the basic framework as shown in Figure 2. Here, the stop-words in Q are removed followed by stemming of the remaining words. The term-frequency values corresponding to the stemmed words are then extracted. The term-frequency table is fed as an input to the SFFS algorithm along with $\mathbb{S}_Q$ and the word-importance scores to find E_Q using the algorithm described in Figure 4.

It is important to stress that all technical choices made in this work that are not our contribution (e.g., using Google as a Web crawler and index and Okapi as a relevance metric), are geared towards simplicity, ease of replicating and rationalizing about our results, and isolation of the effect that the concept of essential pages has with respect to prior work.[4] Virtually every component of our platform could be substantially improved: the ranking metric as a combination of **{C,U,P}** heuristics, the selection algorithm, the conception of the bag-of-words, etc.

[4]For example, using an innovative relevance metric could obfuscate the effect of essential pages on user's learning experience.

4.1 Contextual Noise Reduction

We found in our implementation that for a typical search query, the average number of terms in $\mathbb{S}_Q$ constructed from $N_d^Q = 200$ URLs is approximately 10^4. Closer examination showed that some of these words are unrelated to Q although they receive high-importance scores. We refer to these insignificant words as *contextual noise* (CN). Such words may alter $\mathbb{S}_Q$ significantly and thus, affect the performance of the proposed search engine. In this paper, we propose two techniques for filtering out such words from $\mathbb{S}_Q$.

Paragraph Filtering. We observe that most of the content noise occurs due to blogs and ad aggregators which typically have plenty of information not related to the query such as ads, private communication, off-topic data, etc. To address this issue, we remove all paragraphs across all documents in $\mathbb{D}_Q$ that do not contain at least one of the query terms. We found that such an approach was able to remove significant portion of the contextual noise.

Document Filtering. Motivated by heuristics popular in denoising and in general, pattern classification literature [24], we filter out contextual noise from our results by deriving $\mathbb{S}_Q$ from documents that are not considered for inclusion in E_Q. For our work, we construct $\mathbb{S}_Q$ from documents ranked $1 + N_d^Q/2$ and worse (denoted as the training set) by the providing search engine such as Google. In contrast, we build $\mathbb{D}_Q$ using the top $N_d^Q/2$ documents for rank-ordering. This technique ensures that only words common to both the top $N_d^Q/2$ documents **and** the training documents contribute to the final ranking – as a consequence, significant amount of contextual noise is indirectly filtered from $\mathbb{S}_Q$, i.e., it is not used during the selection process.

4.2 Relevance vs. Coverage

Coverage and relevance are often conflicting requirements in document ranking; they typically result in balancing a trade-off – one of the metrics could improve at the expense of being suboptimal with the other metric. In an ideal scenario, it is important that the search results are relevant to the search query while they are presented with optimized coverage of the knowledge space. To address this issue, we define a combined relevance-coverage metric $\mathcal{RC}(j)$ for each document $j \in \mathbb{D}_Q$ as:

$$\mathcal{RC}(j) = \mathcal{R}(j)^{2(1-\alpha)} \cdot \mathcal{C}(j)^{2\alpha}. \quad (9)$$

Here, $\mathcal{R}(j)$ and $\mathcal{C}(j)$ denote the relevance score and the coverage score for document j, respectively, and are given by Eqs. 2 and 4. The balancing constant, α, could be tailored to specific application requirements. Finally, the new metric, $\mathcal{RC}$, is used in the SFFS algorithm presented as pseudo-code in Figure 4, instead of the $\mathcal{C}$ metric.

5. RESULTS

In this section, we aim to quantify the observed benefits of the proposed technology. We first start by examining a case study in Subsection 5.1. In Subsection 5.2 we present results from a conducted user study with 35 subjects and over 120 queries where we compared our ranking to Google's. Finally, Subsection 5.3 quantifies analytically the obtained coverage and relevance results for a set of benchmark queries.

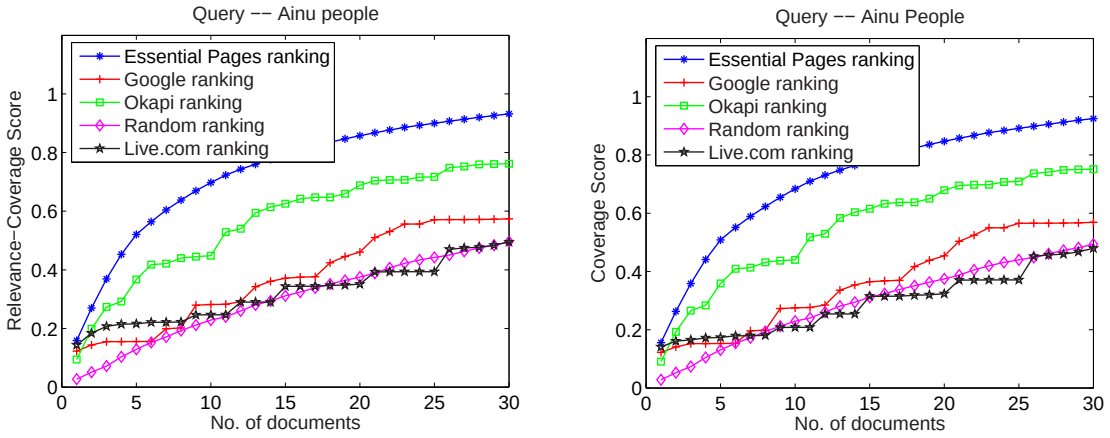

Figure 5: Results for the query `ainu people` showing (a) the relevance-coverage score $\mathcal{RC}(j)$ and (b) the coverage score $\mathcal{C}(j)$ for the top 30 results obtained using each of the five ranking algorithms.

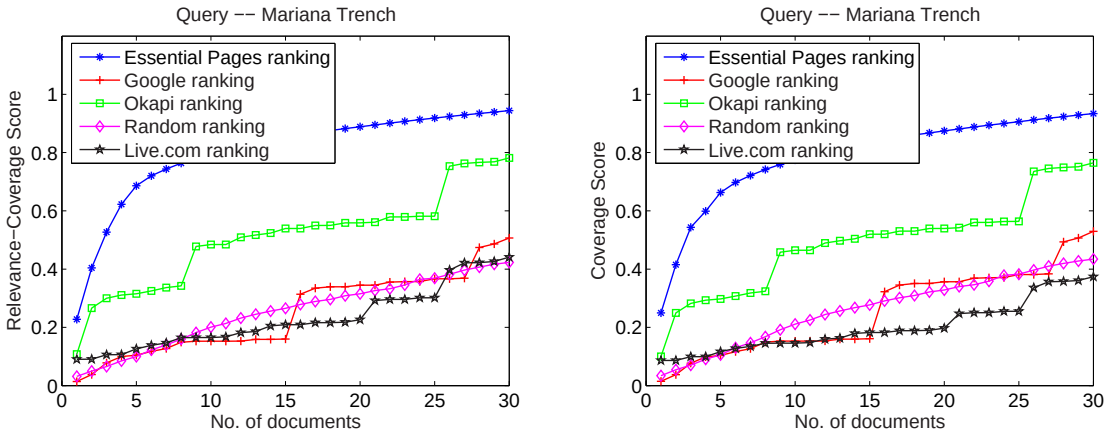

Figure 6: Results for the query `mariana trench` showing (a) the relevance-coverage score $\mathcal{RC}(j)$ and (b) the coverage score $\mathcal{C}(j)$ for the top 30 results obtained using each of the five ranking algorithms.

Table 1: List of most frequent stemmed terms for two exemplary search queries.

Query	`bill gates`		`britney spears`	
Order	Dictionary	Freq.	Dictionary	Freq.
1	gate	3046	britnei	3265
2	bill	2271	spear	2806
3	microsoft	787	music	464
4	busi	354	time	385
5	peopl	325	video	382
6	comput	310	celebr	304
7	softwar	246	album	263
8	job	241	search	254
9	search	235	page	252
10	page	234	pop	246

Table 2: List of most "important" stemmed terms for two exemplary search queries.

Query	`bill gates`		`britney spears`	
Order	Dictionary	Freq.	Dictionary	Freq.
1	live	90	singer	142
2	system	142	live	135
3	product	109	song	149
4	window	193	web	175
5	technolog	128	custodi	204
6	servic	65	peopl	206
7	william	205	mtv	127
8	melinda	141	boi	94
9	internet	108	award	152
10	foundat	195	blackout	165

5.1 Case Study

For our case study, we consider several search queries such as `ainu people`, `alexander mackenzie`, `bill gates`, `britney spears`, and `mariana trench`. Table 1 shows the top 10 most frequent terms in $\mathbb{S}_Q$ for queries `bill gates` and `britney spears`, respectively. We observe that these stemmed terms are highly relevant to the search query. The corresponding stemmed terms that have the highest word-importance scores, $\gamma(w)$, as defined in Eq. 5, are shown in Table 2 for the two queries. For the query `bill gates`, we notice that words such as `live`, `window`, `foundat`, `melinda`

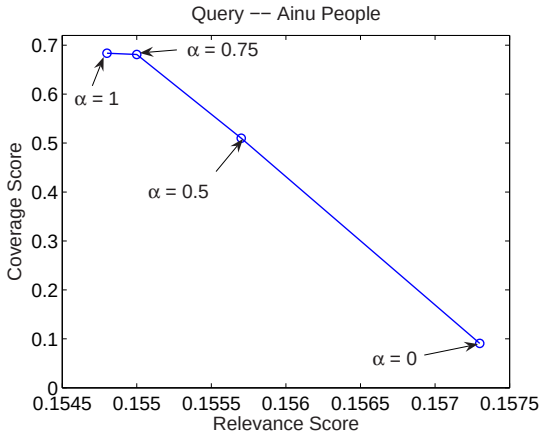

Figure 7: Tradeoff between relevance and coverage for the query `ainu people`.

that are highly relevant to the query and are not covered by all documents, have a high $\gamma(w)$. Similar inferences can be made for the query `britney spears` from Table 2, suggesting that the proposed word-relevance metric can identify important words.

5.1.1 Relevance vs. Coverage

We study the performance of the system as α changes from 0 to 1 in steps of 0.1 and obtain the retrieval results for each step. Consider the query `ainu people`. For each value of α, we record the top 10 URLs obtained from the iterative procedure and compute the joint relevance and coverage score for these pages. In Figure 7, we plot the coverage score with respect to the relevance score for different values of α. We observe from the plot that by reducing α joint relevance increases, while joint coverage decreases. We conclude that with an appropriate choice of α, our system can be tuned to give more importance to relevance or coverage. Most importantly, from the plot one can observe that a small difference in relevance could yield tremendous change in coverage – hence, search engines that are not tailored to address knowledge coverage could suffer from substantial redundancy of their results. In our subsequent experiments, we use $\alpha = 0.5$ for simplicity of computation and to give equal importance to relevance and coverage.

5.1.2 Performance Comparisons

We compare the performance of essential pages with the following state-of-the-art search algorithms:

- **Okapi:** For each document in the database, we compute its Okapi relevance score using Eq. 1. We then rank the documents in $\mathbb{D}_Q$ in decreasing order of their relevance scores to obtain the top 30 documents.
- **Google:** We directly extract the Google ranking for a search query and examine the relevance and coverage of each of the resulting Web-pages. We speculate that Google deploys a proprietary ranking algorithm which evaluates the global link topology [5] as well as other relevance and user behavior models.
- **Live Search:** We also extract the corresponding ranking using Microsoft's Live.com search engine for each search query and examine the relevance and coverage of the resulting Web-pages.[5]
- **Random:** We compare all results to random ranking. To simulate random ranking, we randomly choose 30 documents from the top 100 documents returned by Google and compute their coverage and relevance-coverage scores. We repeat this process 100 times to report the average performance.

For each ranking algorithm, we employ the Okapi metric as given in Eq. 1 to measure relevance and obtain coverage and relevance-coverage scores using Eq. 4 and Eq. 9 with $\alpha = 0.5$. Figures 5 and 6 show the performance results for the considered ranking techniques for search queries: `ainu people` and `mariana trench`, respectively. We observe from the figures that essential pages provide higher search coverage (as shown in Figures 5(b) and 6(b)) and attain better tradeoff between relevance and coverage (as shown in Figures 5(a) and 6(a)) compared to other approaches. The results for the Google ranker are higher than random ranking but lower than the other techniques. This result suggests that while Google returns relevant results, its coverage about the topic is lower compared to other methods. Live.com provides good coverage of the topic in the first few documents but the coverage reduces compared to Google as the number of documents increases. The results for the Okapi ranking algorithm are lower than the proposed algorithm but an improvement over Google and Live.com.

It is important to stress that the analytical results are undeniably skewed to favor Okapi-based rankings – this bias in relevance must be considered when analyzing Figures 5, 6, and 10. First, one way to analytically evaluate the improvement of essential pages is to compare with Okapi-based results only. On the other hand, the data presented for other three reference results, is illustrative of the coverage (but not relevance) that these search engines exhibit. Assuming Okapi performs as poorly as random ranking in true semantic relevance, another way to assess the improvement of essential pages with respect to Google, Live.com, and random ranking, is to consider its coverage score minus the difference between Okapi and random ranking. Most importantly, using both methods essential pages demonstrate substantial improvements over all considered alternatives. Finally, since relevance (see Eq. 1) is slightly correlated to the coverage score (see Eq. 4), they are not completely orthogonal and hence we point to this source of error in performance with respect to true semantic coverage.

5.1.3 An Example Query

We consider the query term `alexander mackenzie` and take a closer look at the retrieved results for this query. It is interesting because there are several celebrities by this name including a person who was the President of Canada between 1873 and 1878, a Scottish-Canadian explorer who is credited for completing the first recorded transcontinental crossing of North America by a European north of Mexico, among others. This query is a learning-type query and we use it to illustrate the proposed methods.

[5]In our experimental results, Live Search is slightly disadvantaged as we have constructed the reference knowledge space, i.e., bags-of-words, for each query using the top 101-200 links obtained from Google. We speculate that this disadvantage is negligent for the final reported results.

Table 3: URLs obtained using essential pages for the query `alexander mackenzie`. The total coverage score obtained using the top 10 links is 0.8846.

S.No	Essential Page Link	Google rank	Page Coverage Score
1	http://www.biographi.ca/EN/ShowBio.asp?BioId=40374	19	0.47
2	http://www.biographi.ca/EN/ShowBio.asp?BioId=36643	20	0.29
3	http://www.bcgrizzlytours.com/mackenzie.htm	82	0.02
4	http://encarta.msn.com/encyclopedia_761563267/Ale..	80	0.16
5	http://www.musicweb-international.com/mackenzie/	30	0.08
6	http://www.bcadventure.com/adventure/explore/cariboo/trails/..	84	0.05
7	http://archives.cbc.ca/IDC-1-73-2320-13530-10/on_this_day/..	77	0.06
8	http://findarticles.com/..	25	0.06
9	http://www.thefreedictionary.com/Sir+Alexander+Mackenzie	28	0.10
10	http://www.fictionwise.com/eBooks/SirAlexanderMackenzieeBooks.htm	61	0.03

The top 10 essential pages are shown in Table 3 along with their corresponding Google ranks and the normalized document coverage score. We observe that all essential pages have a Google rank greater than 18 suggesting that the top 18 Google pages do not provide high information content to cover the topic. The Google ranking even includes the Wikipedia entry for our query among the top ranked links. While pages like Wikipedia are informative, popular, and with high click-through rates, not infrequently they do not provide enough information to substantially cover a topic. For the considered query, the top two essential pages provide most of the information (around 67%) available on-line about the query, while the top two Google pages cover only 3% of the knowledge space, i.e., bag-of-words.

5.2 A User Study

We performed a blind-test user study to comparatively examine the semantic quality of rankings produced by our search engine. Thirty-five people of different age-groups, gender, level of education, and nationality participated in the study. We designed the user interface for our search engine to reflect the demands from the study. On the main search page, the users could enter their query in a text box and hit the search button. The users were asked to particularly focus on learning-type queries for which non-trivial amount of information can be found on the Internet. Also we asked users to refrain from querying content that is covered mainly in technical papers as we did not develop a parser for pdf or postscript file formats, e.g., `graphical models`, `fisher kernels`, `np-hard`.

Once a user would type the query and hit the search button, our program would download the top 200 documents retrieved by either Google or Live.com to a local machine. Our ranker would proceed with the re-ranking as presented in Section 4. We would display the top 10 essential pages alongside the top 10 links retrieved by the resident search engine in the results page as illustrated in Figure 8. The order of their appearance would be randomized (left or right side of the screen) – in addition, the top 10 links were randomly permuted and presented to the user as links "Page 1" through "Page 10" without snippets. The users were asked to carefully browse the content covered by both lists and choose the side (left or right) which, in their opinion, provided better coverage about the search topic. We used voting buttons to record the results. The users had three options: left side better than the right, both equal, and right side better than the left. This was done with an objective to obfuscate the source of the two top 10 lists. Each user was asked to post three or four queries as the user study took around half hour to one hour per subject.

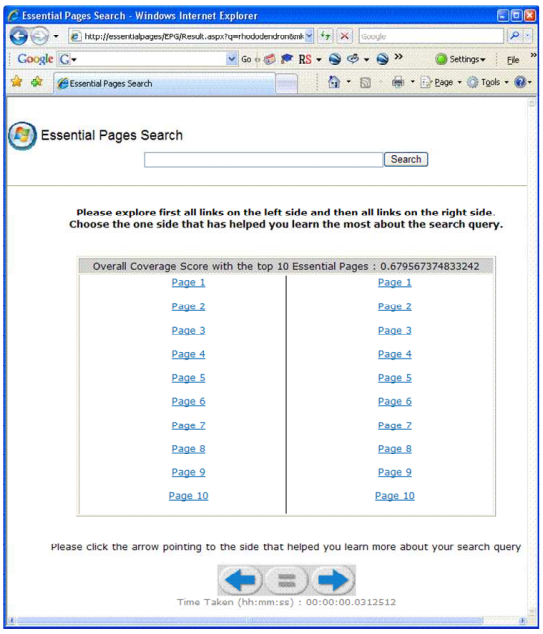

Figure 8: Screenshot showing the results page in the user study.

The results from the user study are shown in Table 4. The top subtable shows 80 queries for which users chose the essential pages over Google; in the middle are the 18 queries for which users chose equal; and the bottom has the list of 22 queries for which Google results were preferred. This suggests a 5:1 ratio of queries where essential pages were equal or better than Google. The results for one query, `rene magritte`, were selected by two different users as "equal" and "Google better." This query was one of five queries given to users as examples of a learning-type of query. All other queries in Table 4 were "invented" by users during the user study. We filtered from our final results reported in Table 4 only the queries that equaled participants' names.

These results demonstrate the potential of the proposed technique for learning-type queries, in particular as we did not use directly any other ranking criteria from the **{C,U,P}** set. In this paper, we do not propose any hybrid ranking schemes mainly to retain a focus on providing an insight into how important is to remove redundancy from search results and present pages that jointly cover as much knowledge about a query as possible. Clearly, hybrid ranking schemes could demonstrate the benefits of all contributing technologies and thus improve user satisfaction even further.

Table 4: Queries from the user study.

Queries for which users picked essential pages over Google (80 in total)
aperture mode, bill gates, tiger shark, koh phi phi, the pistol star, paul gascoigne, george best, paris hilton, mariana trench, mount rainier, click fraud, beckenbauer, alexander mackenzie, instant messaging, hiroshima bombing, hunters of dune, manmohan singh, gottfried von leibniz, beijing summer olympics, bourne ultimatum, breastfeeding, frida kahlo, galapagos, microsoft, tips on photography, alexander hamilton, mahatma gandhi, indian music, berlin wall, roger federer, machine learning, how cell phones work, the office tv series, provo, turkmenistan president, izmir places to see, yellow stone national park, how to write explorer bho, tour de france, cancun, microsoft, mt baker, jimi hendrix, the space needle, interwinding capacitance, electronic bandgap filter, mykonos greece, czech republic, microsoft, search engines, halo, golden retriever, tiberius bird flu, countrywide, composition roof, acoustic source localization, chinese food, smartphone, starcraft, fishing industry in the maritimes, academy awards, cn tower, carcinogens in artificial sweeteners, nikon d200, fed repo rate, mfa arbitrage, german shepherd breed children, rhododendron, adjusting aperture and shutter speed to take good photographs, catherine zeta jones, social psychology, marathons, edmund burke, anarchism, summer internship, northern light, denali national park, new york yankees, acute mountain sickness, randy johnson
Queries for which users decided that essential pages technique and Google were equal (18 in total)
kestrel, douglas engelbart, *rene magritte*, jay leno, turbine in jet engine, bando, the great red spot, planet earth, space elevator, cheapest car stereo, most fuel efficient car, ainu people, yellow stone volcano, cubs, carl gauss, fed interest rate, subprime mortgage, bharatnatyam
Queries for which users choose Google results over essential pages (22 in total)
format string attack, borgia, quorum systems, the fidelius charm, paul mccartney, bora milutinovic, seth green, ultimate frisbee, nonparametric statistics, racial profiling, price jigme, saint bernard dog, gibson magic guitar, mlb and steroids, microscopy, ian anderson, *rene magritte*, weeds tv series, sweating sickness, gdansk poland, martial arts, how to play poker

5.3 Discussion

In this subsection, we further analyze the efficiency of the proposed technique in terms of the performance metrics introduced earlier in the paper. For this study, we employed the 120 queries obtained from the user study. In addition, we included another set of 60 learning-type queries from Google's top 100 most popular queries.

5.3.1 Coverage

We examine the coverage results over our benchmark in terms of the metric introduced in Eq. 4. Figure 10 shows the performance of the proposed techniques, averaged over the 180 search queries, and compares the results with the other schemes as discussed in Section 5.1. We notice from Figures 10(a) and 10(b) that the proposed scheme provides significantly higher relevance-coverage and coverage results. Further, the figure shows that the top 30 essential pages, on average, jointly cover almost 95% of the total information content available (modeled as a bag-of-words) on the Web pertaining to the search query. This result suggests that if learning about a topic is the main objective behind a query, then essential pages would reduce the required time for browsing the Web to access the required information. In comparison, other ranking algorithms did not perform as well. Popular search engines like Google and Live Search

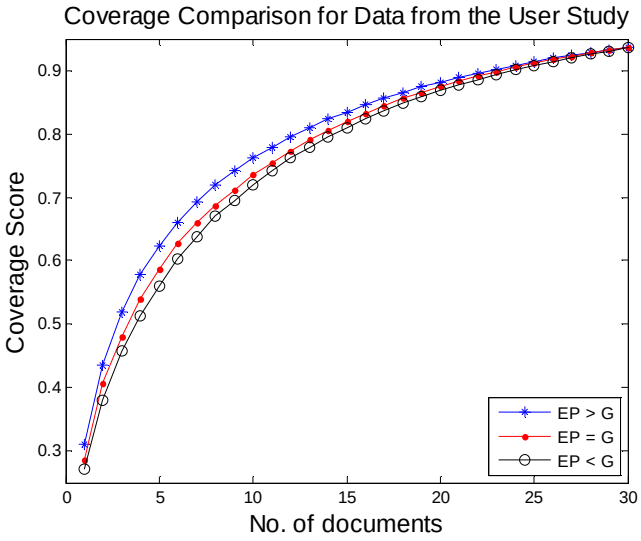

Figure 9: Are our analytical results semantically meaningful? Illustration of the average coverage for essential pages created for queries where essential pages were identified as better ($EP > G$), equal ($EP = G$), or worse ($EP < G$) than Google.

only provided around 58% and 45%, respectively, of the total information content in the top 30 links, and Okapi performed slightly better with 70% coverage. These results also supplement the 5:1 essential pages over Google preference ratio obtained from the user study.

We attempt to connect the response from the user study from Table 4 with our analytical results in Figure 10. We independently illustrate the average coverage for essential pages created for queries where essential pages were identified as better, equal, or worse than Google in Figure 9. First, strong separation of the analytical results for the semantically separated data showcases that our analytical results are semantically meaningful. Next, one can observe that the coverage for queries where essential pages were identified as better than Google is substantially higher in particular for the first 10-15 documents – then, as the number of documents increases towards 30 this advantage disappears. Since the number of considered documents was 100, we speculate that with larger $\mathbb{D}_Q$, one could attain better use of essential pages as it would present only relevant and uncovered documents to the end user.

5.3.2 Relevance

Essential pages not only provide good coverage, but also represent relevant Web-links. Figure 10(c) shows the average relevance, measured using the Okapi metric in Eq. 1, of the top 30 essential pages in comparison with other ranking schemes. As expected the Okapi based ranking technique retrieves documents with the highest relevance. Essential pages come second in the comparison. Referring back to Figure 7, we remark that the proposed technique can attain a better trade-off point with much improved coverage at a cost of slightly lower relevance of the retrieved results.

We also support the hypothesis that coverage and relevance are highly mutually independent. We placed each considered document from our benchmark of 180 queries

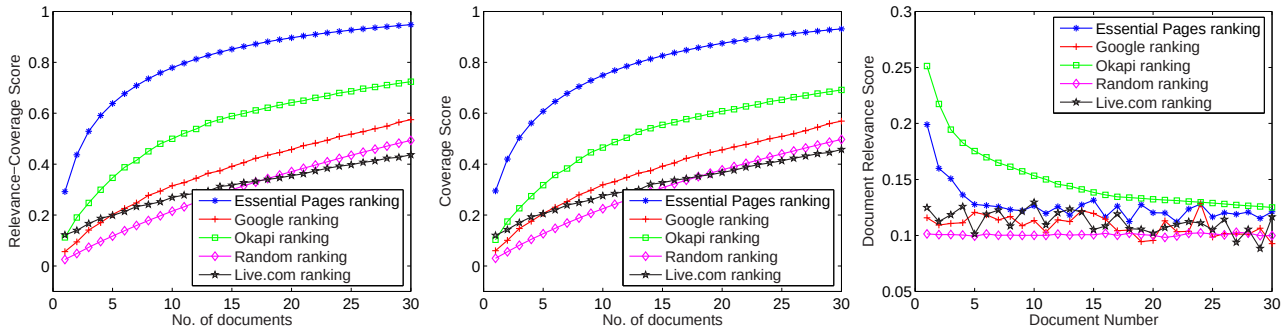

Figure 10: Average performance of the essential pages technique compared with existing ranking approaches for the 180 queries obtained in the user study. The figure shows the performance in terms of: (a) relevance-coverage, (b) coverage, and (c) relevance of each document returned.

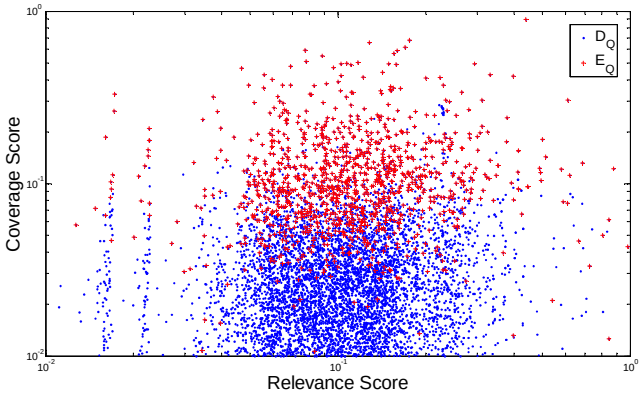

Figure 11: Illustration of the dependency of coverage and relevance in the set of queries presented in Table 4. The plot presents the value of the coverage and relevance for each document in $\mathbb{D}_Q$ and each essential page in E_Q for each query Q in the set.

(100 Web-pages per query) in a relevance-coverage coordinate system in Figure 11. Pages that have been selected as essential are marked with crosses. One can observe that although in our definitions for both relevance and coverage there is undeniable correlation (see Eqs. 1 and 4), our platform selected documents that favored coverage but not at the expense of redundancy and seemingly independently of document relevance. On the other hand, the underlying search engine returned results that are mostly relevant to the query; one could argue that our platform could almost not "miss" on the relevance – however, we note that our system was capable of ranking the top results better in most cases by analyzing the nuances in the mutual dependence of their coverage-relevance scores. Thus, we suggest that our system would perform best when applied to an existing search engine as a post-processing tool.

5.3.3 Comparison to Google

In this part, we take a closer look at the Google ranks of the top 30 essential pages. Figures 12(a) and 12(b) show the percentage of search queries with a particular Google rank for the first and the fifth essential page, respectively. These histograms show that the Google ranks for these pages vary from 1 to 100 almost uniformly. This result suggests that the pages with maximum coverage of the topic are not necessarily in the top of Google results; and that users have to browse through many Google result pages to finally reach a link that introduces significant new content. In Figure 12(c), we show the histogram of the difference between the Google rank and the essential page rank, for the top 10 links and 180 queries. These results supplement our results in Figure 12(a) and (b), and demonstrate that the top essential pages could appear anywhere in the top 100 Google pages almost uniformly.

5.3.4 Computational Complexity

Finally, we examine the computational complexity of the proposed technique. Here we must stress that our implementation did not focus on engineering challenges hence all provided results are marked as far from optimal. Figure 13 shows the actual run-times for the different processes involved in our implementation of essential pages over 100 runs for different queries. Starting from a downloaded set of documents the total time taken for identifying the essential pages was around 4 seconds out of which the SFFS algorithm took less than 2 seconds. The latter number is exceptionally important ("slow") as this step is the only one that needs to be executed in run-time for a given query unless the result is already cached. Typically, there would be pressure to complete the run-time task within 10-100 milliseconds for a specific query – hence the question whether essential pages could be identified 20-200 times faster than the run-times we are reporting. Since the main culprit to current performance is the computation of joint coverage (computed *ex novo* for each step in the selection algorithm), we speculate that simple heuristics such as greedy set covering [25] would provide desired run-times at little cost to ranking quality.

5.3.5 Extensions to the Proposed Scheme

We proposed a novel metric for measuring coverage based on cumulative term-frequency data from a set of documents and employed the Okapi metric in Eq. 1 to measure relevance of search results. We chose the Okapi metric for two reasons: *a*) it has been shown to be an effective metric for measuring document relevance [19], and *b*) it has a simple closed-form expression that facilitates analysis. The proposed techniques are not limited to the Okapi metric; other metrics from the **{C,U,P}** set can be employed as well.

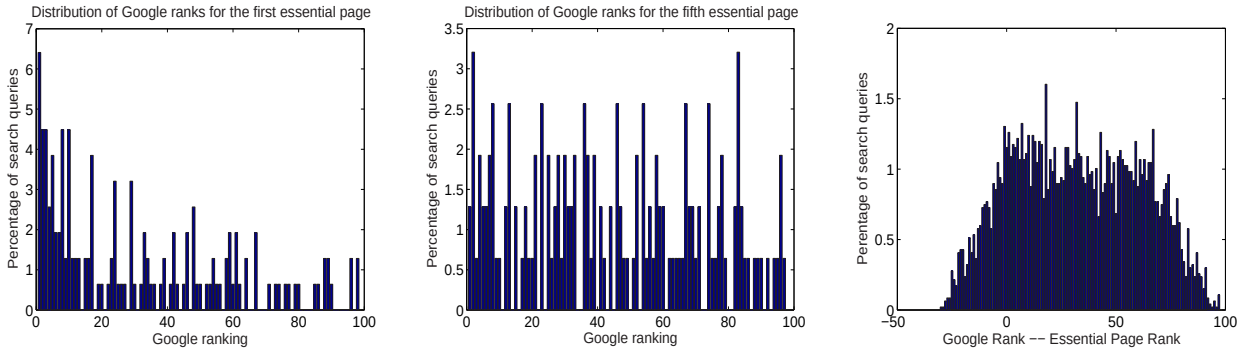

Figure 12: Distribution of Google ranks for the (a) first and (b) fifth essential page; (c) shows the histogram of the difference between Google ranks and the essential page ranks.

We bi-partition existing relevance metrics into two classes: analytical and user-generated. The analytical class bets on a set of non-semantic features of a document or its position in the global topology of related Web-pages to set its relevance. One such example is Lucene which supports scoring functions based on the Vector Space Model [26].

User-generated metrics include various forms of semantic filtering for document relevance. For instance, metrics such as recall (a measure of the ability of a system to present all relevant items), precision (a measure of the ability of a system to present only relevant items), and precision@10, may be used to measure relevance [27]. Another popular metric is the normalized discounted cumulative gain (NDCG) [28, 29], that has been shown to give more credit to systems with high precision at the top ranks than other evaluation measures [28]. In an alternate implementation, we would build essential pages by replacing the relevance metric, $\mathcal{R}(j)$, in Eq. 9 with document's NDCG-based ranker and retaining the remainder of the ranking algorithm intact. As a disadvantage, NDCG-based rankers suffer from imperfections of human filters: conquering the full extent of the knowledge space for a specific query is a cumbersome task – hence the judgements of relevance in common benchmarks are arguable at best [28].

Nevertheless, we emphasize that the full potential of essential pages can be explored only in hybrid schemes where a subset of **{C,U,P}** metrics is considered to provide qualitative benefits to building a relevant knowledge space for a specific query. The value of the work presented in this paper is in the demonstration that simple Okapi-relevance-coverage metric can provide substantial improvements in serving Web-search results over state-of-the-art search engines both analytically and from users' perspective.

5.4 Detecting a Learning-Type Query

Essential pages are best served to users when applied to a relatively small set of documents returned by an off-the-shelf search engine as the most relevant, popular, and/or user-acknowledged. Essential pages also improve search results only for learning-type queries. With the exception of encyclopedia-style Web services where all queries are of learning-type, identifying such a query automatically at the server or client in the general case could be a challenging task. Possible simple heuristics are:

- depth of click-throughs – the deeper, the more likely the query is of learning-type;
- query terms present as an encyclopedia entry,
- number of documents relevant to the query found on the Web – we found empirically that queries that have exceptionally large ($> 10^7$ documents) or small presence ($< 10^4$ documents) on the Web are not learning-type queries.
- absence of multimedia in a large subset of the resulting pages – learning-type queries are typically dominated by textual results, even in the case when the topic is associated with visual effects (i.e., painter biography, computer graphics algorithms).

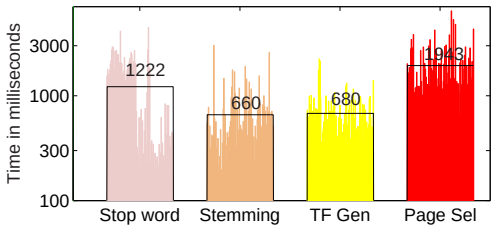

Figure 13: Run-times for individual steps of the essential pages ranking algorithm. Results were obtained using a 3.4GHz Pentium with 2GB RAM.

We stress that the software that selects essential pages could be placed at the server as well as pushed to the client. In the latter case, one way of serving the results in response to a user query is to present the response from the underlying search engine first, then, download resulting HTML files without multimedia, and re-rank them as the user browses the resulting list of links.

Acknowledgments

We would like to thank Chris J.C. Burges, Krysta Svore, and Jaime Teevan from Microsoft Research and Yisong Yue from Cornell University for suggestions and comments that have improved the content of this manuscript.

6. SUMMARY

State-of-the-art Web-page ranking is a well researched multi-dimensional search process that still leaves a lot to be desired. In this paper, we introduce a novel important dimension in this space: coverage of the query topic. Essential pages are constructed to provide high coverage of the knowledge related to the query as a set, rather than individually. As we have modeled a knowledge space using a bag-of-words, the novel ranking problem can be formulated as finding a subset of pages of given cardinality that covers as many as possible terms in the bag. The resulting rankings, when compared to state-of-the-art, show substantial differences. Analytically, essential pages show significant improvements over existing methods; in a simple user study, essential pages were selected 2:1 as better than, and 5:1 as better or equal to Google rankings. We stress that the obtained results were created using only a simple coverage-relevance criterion for ranking. We anticipate that hybrid ranking algorithms that take into account our objective can reach even greater user approval.

7. REFERENCES

[1] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. Proc. of the WWW, pp.107–117, 1998.

[2] S.E. Robertson and K. Sparck-Jones. Relevance weighting of search terms. Document Retrieval Systems, Vol.3, pp.143–60, 1988.

[3] S. Lawrence and L. Giles. Context and page analysis for improved Web search. IEEE Internet Computing, vol.2, no.4, pp.38–46, 1998.

[4] E. Agichtein, et al. Improving Web Search Ranking by Incorporating User Behavior. ACM SIGIR, 2006.

[5] L. Page. Method for scoring documents in a linked database. US Patent no. 6,799,176, 2004.

[6] L. Page. Method for node ranking in a linked database. US Patent no. 7,058,628, 2006.

[7] A.N. Langville and C.D. Meyer. Deeper Inside PageRank. Internet Mathematics, Vol.1, (no.3), pp.335–80, 2003.

[8] J. Kleinberg. Authoritative sources in a hyperlinked environment. ACM SODA, 1998.

[9] K. Collins-Thompson and J. Callan. A language modeling approach to predicting reading difficulty. HLT/NAACL, 2004.

[10] D. Harman. Overview of the TREC 2002 novelty track. TREC, 2003.

[11] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to Web search results. Computer Networks, Vol.31, pp.1361–74, 1999.

[12] A. Broder. A Taxonomy of Web Search. SIGIR Forum, Vol.36, (no.2), 2002.

[13] D.E. Rose and D. Levinson. Understanding User Goals in Web Search. WWW, 2004.

[14] P. Pudil, et al. Floating search methods in feature selection. Pattern Recognition Letters, vol.15, no.11, pp.1119–25, 1994.

[15] H. Chen and D.R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. ACM SIGIR, 2006.

[16] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. ACM SIGIR, pp.335–336, 1998.

[17] C. Zhai, et al. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. ACM SIGIR, pp.10–17, 2003.

[18] B. Zhang, et al. Improving Web Search Results Using Affinity Graph. ACM SIGIR, 2005.

[19] S.E. Robertson and K.S. Jones. Simple Proven Approaches to Text Retrieval. Tech. Report TR356, Cambridge University Computer Laboratory, 1997.

[20] W. B. Frakes and R. Baeza-Yates. Information Retrieval: Data Structures and Algorithms. Prentice Hall, 1992.

[21] C.J. van Rijsbergen, et al. New models in probabilistic information retrieval. London British Library R&D Report, no.5587, 1980.

[22] P. Schauble. Multimedia Information Retrieval: Content-Based Information Retrieval from Large Text and Audio Databases. Springer, 1997.

[23] List of stop words, available on-line at: `http://www.dcs.gla.ac.uk/idom/ir_resources`.

[24] R.O. Duda, et al. Pattern Classification. John Wiley & Sons, Inc., Second Edition, 2000.

[25] T.H. Cormen, et al. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001. Section 35.3, The set-covering problem, pp.1033–8.

[26] R.A. Baeza-Yates and B.A. Ribeiro-Neto. Modern Information Retrieval. Addison Wesley, 1999.

[27] D.K. Harman. Common Evaluation Measures. In Appendix, Proceedings of Text Retrieval Conference, 2005. Available on-line at: `http://trec.nist.gov/`.

[28] K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. ACM SIGIR, pp.41–48, 2000.

[29] I. Matveeva, et al. High Accuracy Retrieval with Multiple Nested Ranker. ACM SIGIR, 2006.