Similarity Search using Concept Graphs

Rakesh Agrawal Sreenivas Gollapudi Anitha Kannan Krishnaram Kenthapadi
Microsoft Research
Mountain View, CA, USA
{rakesha, sreenig, ankannan, krisken}@microsoft.com

ABSTRACT

The rapid proliferation of hand-held devices has led to the development of rich, interactive and immersive applications, such as e-readers for electronic books. These applications motivate retrieval systems that can implicitly satisfy any information need of the reader by exploiting the context of the user's interactions. Such retrieval systems differ from traditional search engines in that the queries constructed using the context are typically complex objects (including the document and its structure).

In this paper, we develop an efficient retrieval system, only assuming an *oracle* access to a traditional search engine that admits 'succinct' keyword queries for retrieving objects of a desired media type. As part of query generation, we first map the complex query object to a *concept graph* and then use the concepts along with their relationships in the graph to compute a *small* set of keyword queries to the search engine. Next, as part of the result generation, we aggregate the results of these queries to identify relevant web content of the desired type, thereby eliminating the need for explicitly computing similarity between the query object and all web content. We present a theoretical analysis of our approach and carry out a detailed empirical evaluation to show the practicality of the approach for the task of augmenting electronic documents with high quality videos from the web.

1. INTRODUCTION

The rapid growth in the availability of online content in various formats such as images, videos, and podcasts, and the growing popularity of mobile devices with rich interactive applications engage the user with an immersive and interactive experience. Such an experience often gives raise to implicit information need, calling for new retrieval systems that can retrieve results in response to a complex query object such as a document or a video. For example, consider a scenario in which a tablet application automatically satisfies the information need of a user by finding relevant media from the web implicitly based on her current interactions with the application, that is, without the user having to resort to a search session.

Two problems come to the fore in the above settings, *viz.*, imputing the context and defining an effective search paradigm using the context. While there has been work on developing query / ranking models that exploit context [3, 6, 11, 20, 33], the next logical step of implicitly getting the information to the user *without* a search query is recently gaining attention in the research community [16]. In fact, in the case of mobile search, there has been research to incorporate context such as the user's location into the search and return location-specific results [2, 27, 39]. However, in scenarios such as e-reader application that we described earlier, the context can include information in the page, and the pages visited by the user. Such context can result in a complex object representation that can not be handled by the existing search engines.

We focus on the following problem: *Given a document that also specifies the context, how do we retrieve web objects that are relevant to it?*

One solution to this aforementioned problem is to define an effective representation for the given document (query) as well as the web content (results) and a notion of similarity that captures relevance between a query and a result. Both these components are challenging due to various logistical and efficiency reasons. For example, it is non-trivial to design an index that can perform efficient retrieval at scale. In addition, it is not always possible to have an explicit unified representation and even when such a representation exists, it may be computationally prohibitive to perform similarity computations over the representation.

In this paper, we propose an alternative approach that only assumes access to a search engine that admits 'succinct' keyword queries for retrieving objects of desired type. Our approach defines a universal concept graph across all web objects, while, in practice, it is realized *only* for the query object and thus does not require that the query and the results share the same representation. Then, it uses the relationships embedded in the concept graph to identify a *small number* of keyword queries consisting of combinations of these phrases which are then queried on the search engine. The results of these queries are aggregated to identify the relevant web content, thereby eliminating the need for explicitly defining a similarity function. We also formally provide theoretical guarantees for the algorithm.

In order to evaluate the system, we mimicked the e-reader application where the goal is to automatically enrich each textbook section (wherein the context is the section being

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM'14, November 3–7, 2014, Shanghai, China.
Copyright 2014 ACM 978-1-4503-2598-1/14/11 ...$15.00.
http://dx.doi.org/10.1145/2661829.2661995.

currently read) with web content (such as videos, images, and web articles). Our extensive experiments over a corpus of textbook sections spanning multiple subjects demonstrate the efficacy of our system.

2. RELATED WORK

The related work can be broadly classified into four areas of research, *viz.*, contextual search using new query models and/or ranking models that incorporate context of the user query; query session analysis wherein the focus is on understanding a user's search session using click feedback as well as relationships between queries; finding succinct representations for both queries and documents towards admitting efficient similarity search in the underlying high dimensional space; and finally the notion of the aboutness of a document.

There has been a lot of research in the area of contextual search where the query is implicitly (without the user's input) "expanded" to include the context in which the query is issued. Commonly studied set of contexts include surrounding material around a user query in a document, a user's location and other local information such as weather [2, 3, 6, 11, 20, 27, 32, 33, 39]. Our work differs from this line of work in that it considers more complex context arising out of user behavior in rich mobile applications such as e-reader, where the context is larger than a few keywords.

One can draw a similarity between our query model in which we "map" the concept graph representation of a document to an algorithmically selected set of concept phrase pairs, and a query session of a user in which she may issue multiple (often related) queries to the search engine [4, 10, 14, 15, 25]. The key difference arises in the fact that in our model, the mapping is algorithmically achieved using the context (such as the information on the page the user is reading) while in the case of query sessions, the analysis is performed by mining the query logs containing queries explicitly issued by the user.

Another related area of work is semantic search which focuses on relationships between the query terms in addition to the query terms as part of the query. The relationships are often encoded using succinct data structures (e.g., concept graphs) [38].

A key component of a similarity search solution is an effective representation of the query and the document objects, which has been studied extensively [8, 9, 35, 37]. For example, techniques for indexing graph databases and efficient processing of graph queries have been proposed in [35, 37]. Another line of closely related work is querying using more complex representations beyond a small number of keywords. Recent research includes the query by document work [36] that extracts key phrases from a document and uses them as queries to search for similar documents (see references therein), and information retrieval with long queries which focuses on either selecting a subset of the query, or weighting the terms of the query [34]. This body of research however is not designed to work for queries over concept graphs which also encode relationship between the phrases.

In another line of related work, a central question often studied is the "aboutness" of a document with the goal of effectively summarizing the information in a document, which can then enable efficient similarity search using the document as a query. This notion has been been extensively investigated in the information retrieval literature from both theoretical (*e.g.*, [5, 17, 19]) as well as pragmatic perspectives (*e.g.*, [21, 23, 26]). Similarly, multiple alternatives exist for algorithmically extracting concept phrases from a text ranging from detecting key phrases based on rules (grammar) to statistical and learning methods. In the former, the structural properties of phrases form the basis for the rule generation. In the latter, the importance of a phrase is computed based on statistical properties of the phrase [21].

3. MODEL

3.1 Representation of Documents using Concept Graphs

Our model represents a given document as well as all web content as part of a universal concept graph whose nodes correspond to a universe of concept phrases and edges correspond to relationship between them. This representation enables similarity search across different media types, for example, a web video for a text document, or a web article for a video. Hereafter, we use the term 'document' to refer to an object of any media type that can be represented as part of the universal concept graph.

Formally, let U denote the set of all concept phrases. We represent each document with respect to the universal concept graph, $\mathcal{G} = (U, E)$ defined over the set of all concept phrases. Edges in $\mathcal{G}$ denote relationship between concept phrases, that is, $(c_1, c_2) \in E$ if and only if c_1 and c_2 are related to each other. Given any subset of concept phrases $X \subseteq U$, let G_X denote the subgraph of $\mathcal{G}$ induced by X, that is, G_X is defined over X and consists of all relationship edges between concept phrases in X.

For a given document s, let $C(s) \subseteq U$ denote the set of concept phrases present in s. We also define a set of *core concept phrases* associated with document s, and denote it by $\Gamma(s)$. We require that each core concept phrase is present in U (that is, $\Gamma(s) \subseteq U$), but not necessarily present in s (that is, $\Gamma(s)$ need not be a subset of $C(s)$). Denote the union $C(s) \cup \Gamma(s)$ by $\Pi(s)$. We further include the set of relationship edges between phrases in $\Pi(s)$ as part of our representation of s. Thus, we represent s as the triplet $(C(s), \Gamma(s), G_{\Pi(s)})$.

We emphasize that while $C(s)$ corresponds to concept phrases occurring in s, $\Gamma(s)$ corresponds to a small set of core concept phrases that capture the focus or the essence of s. Intuitively, a core concept phrase should be related to several concept phrases occurring in s. However, we do not require a core concept phrase to always occur in s, thereby enabling richer representations.

We remark that our representation can also be applied to web content of a desired media type assuming that the corresponding set of concept phrases can be determined. For example, the concept phrases corresponding to a video or an image from the web could be obtained through associated metadata such as title, caption and other descriptive text, transcripts, tags, associated anchor text and queries. Given the set of concept phrases, our algorithm for computing the set of core concept phrases can then be applied to such documents. Our model can be thought of as a generalization of the representation of documents as points in high dimensional space, where dimensions correspond to concept phrases. Our model reduces to the latter when the relationship graph has no edges. In other words, our model takes into account dependencies between concept phrases since the dimensions are typically not independent of each other.

3.2 Similarity Measure

Our goal is to determine web content of a desired media type that is most relevant or similar to the given document. Since web content can also be thought of as being represented with respect to the universal concept graph, we state the desiderata for similarity in terms of the above representation. A web document that overlaps completely with the given document with respect to both the set of core concept phrases and the set of concept phrases has the largest similarity. Since the core concept phrases are intended to capture the focus of the document, our similarity measure places greater emphasis on overlap with respect to the set of core concept phrases compared to the set of concept phrases. Further, when these sets do not completely overlap, there is greater similarity if the corresponding induced graphs are "near" each other. We compute the extent to which a core concept phrase and the phrases in its relationship neighborhood are common across the two documents, and aggregate over all core common phrases in the given document to obtain the similarity measure. In this manner, to be considered very similar, a web document needs to have large commonality with the focus or the essence of the given document, and further the web document needs to substantially share the concept phrases that are discussed in the context of the focus of the document.

Formally, we define the similarity of a web document d with respect to s as:

$$\text{SIM}_s(d) = \sum_{\tilde{c} \in \Gamma(s)} sim_{\mathcal{G},\tilde{c}}(\Pi(s), \Pi(d)),$$

where $sim_{\mathcal{G},\tilde{c}}(X, Y)$ represents a measure of similarity between sets of concept phrases X and Y in the context of the universal concept graph $\mathcal{G}$ and a core concept phrase $\tilde{c}$. For example, we can define $sim_{\mathcal{G},\tilde{c}}(X, Y) := \beta_{\tilde{c}}(G_{local,X}(\tilde{c}), G_{local,Y}(\tilde{c}))$, where $G_{local,X}(\tilde{c})$ represents a subgraph of G_X consisting of $\tilde{c}$ and its neighborhood, and β represents a measure of similarity between two graphs. These subgraphs can be defined by expressing the neighborhood in terms of the concept phrases in $\mathcal{G}$ reachable within a certain number of hops from $\tilde{c}$. Let $B(\tilde{c}, r)$ denote the ball of radius r around the concept phrase $\tilde{c}$, that is, the set of concept phrases that are reachable within r hops from $\tilde{c}$ in $\mathcal{G}$. We can then define $G_{local,X}(\tilde{c}) := G_{B(\tilde{c},r) \cap X}$ (that is, the induced graph formed by concept phrases present in X that are reachable within a small number of relationship edges from $\tilde{c}$). The measure of similarity between two graphs (β) admits different choices, for example, the size of the maximum common subgraph.

Similarity between concept phrases: We also associate a notion of similarity $sim(c_1, c_2) \in [0, 1]$ between two concept phrases c_1 and c_2 in the context of $\mathcal{G}$. It can be viewed as the similarity between sets $\{c_1\}$ and $\{c_2\}$ in the context of $\mathcal{G}$, $sim_{\mathcal{G}}(\{c_1\}, \{c_2\})$. We use the notation $sim(c_1, c_2)$ for brevity. We can compute this similarity in different ways, for example, as Jaccard coefficient between the neighborhood sets around c_1 and c_2: $sim(c_1, c_2) := \frac{|B(c_1,r) \cap B(c_2,r)|}{|B(c_1,r) \cup B(c_2,r)|}$.

3.3 Oracle Model for Computing Similarity

Since we have a rich representation for the given document but do not have a corresponding rich, explicit representation for web documents, we compute the above similarity measure by making use of a suitable oracle, namely a search engine, and performing search engine probes in a lower dimensional space.

Algorithm 1 CGSimilarity

Input: A document s; Number of desired similar web documents m; Search engine SE to retrieve web documents of a desired media type; Universal concept graph $\mathcal{G}$.
Output: A list of m web documents most similar to document s.

1: Identify a set $C(s)$ of (up to) top n concept phrases from document s using $\mathcal{G}$ (§4.1).
2: Identify a set $\Gamma(s)$ of (up to) top γ core concept phrases associated with document s using $\mathcal{G}$ (§4.2).
3: Perform search engine probes in a lower dimensional space: Form queries by selectively combining core concept phrases in $\Gamma(s)$ with concept phrases in $\Pi(s)$ that are within the relationship neighborhood. Obtain (up to) top t results for each of the queries from search engine SE (§4.3).
4: Aggregate over the result lists to obtain the relevance score, $\nu_s(d)$ for each web result d, and return top m web results (§4.3).

4. ALGORITHMIC APPROACH

We next present our algorithm for obtaining web documents of a specified media type that are similar to a given document. Algorithm CGSimilarity takes a given document as input, and returns a specified number of most similar web documents. The algorithm first identifies the set of concept phrases as well as the set of core concept phrases in the document with respect to the universal concept graph. Then, the algorithm forms queries by selectively combining concept phrases associated with the document, issues these queries to a search engine for retrieving documents of a desired media type, and aggregates the search engine results to identify relevant web content. We describe each of these components below.

We remark that our algorithm can be viewed as a broad framework that makes use of a rich document representation and a web-scale search engine in order to retrieve similar web documents. We obtain different algorithmic instantiations through different choices of the components. For instance, Comity algorithm [1] corresponds to the instantiation where the document is represented as a set of concept phrases, and all combinations of concept phrases are issued as queries. In this paper, our focus is on providing a richer representation of the document as part of the universal concept graph, and using this representation to identify a small set of core concept phrases (step 2), and thereby selectively form queries (step 3).

Universal concept graph: Building upon [12, 24, 30], we define the set of all concept phrases, U to be the set of all Wikipedia article titles. The relationship between two concept phrases is obtained using Wikipedia linkage structure, that is, an (undirected) edge (c_1, c_2) is included in the universal concept graph $\mathcal{G}$ if the corresponding Wikipedia articles mutually hyperlink to each other.

4.1 Identification of Concept Phrases

We next present the algorithm used in our implementation for identifying concept phrases from a *text document* (Algorithm ConceptPhraseIdentification). We emphasize that our model as well as algorithmic approach apply to documents of other media types as well, assuming that the set of concept phrases in the document is provided by an oracle. Given the set $C(s)$ of concept phrases associated with document s, the remaining components of Algorithm CGSimilarity can be applied in the current form.

Multiple alternatives exist for identifying concept phrases from a text [21, 23]. Our implementation defines the initial set of concept phrases to be the phrases in s that are significant *k-grams*. Significant unigrams, bigrams, and trigrams, along with their associated significance scores, are determined as described below. The concept phrases relevant to our application typically consist of terminological noun phrases containing adjectives, nouns, and sometimes prepositions [22]. A concept phrase is unlikely to contain other parts of speech such as verbs, adverbs, or conjunctions. Hence, we retain only phrases that are terminological noun phrases (using the linguistic pattern A^*N^+, where A is an adjective and N a noun). Examples of phrases satisfying this pattern include 'probability density function', 'economic policy', and 'mechanical energy'. Next, we retain those phrases that map to Wikipedia article titles. In this manner, we ensure that the set of concept phrases obtained from the document is indeed a subset of the set of all concept phrases, U defined earlier. We then form a directed graph H induced by Wikipedia linkage structure over these phrases, and propagate the scores through H in each of r iterations. In each iteration, the score for a phrase c is divided equally among the phrases that are hyperlinked from (and hence endorsed by) the article with title c, and the new score for c is determined by summing the values received from phrases (articles) that hyperlink to c. Finally, we select the top phrases based on these smoothed scores.

Extraction of significant k-grams: We first generate all possible unigrams (single words), bigrams (two contiguous words), and trigrams (three contiguous words) from s. For $1 \leq k \leq 3$, a k-gram r is associated with a score $\alpha(r) = f(r)/N_k$ where $f(r)$ is its frequency and N_k is the total number of occurrences of all k-grams in s. We prune k-grams with score below a certain minimum threshold. Next, for $k \in \{2, 3\}$, we define the significance of a k-gram in terms of the relative likelihood of the underlying words appearing together (inspired by generative language model literature, for example, for query segmentation [31]) , and for $k = 1$, the significance score is set to be the initial score above. In particular, the significance of a bigram AB is computed as $\delta(AB) := \frac{\alpha(AB)}{\alpha(A)\cdot\alpha(B)}$, and those below a threshold are pruned. For example, 'San Francisco' is likely to be considered a significant bigram since most occurrences of 'Francisco' are likely to be preceded by 'San', and a large number of occurrences of 'San' are likely to be followed by 'Francisco'. This computation can be extended to a trigram ABC by considering two possible ways to split ABC into two parts, and taking the larger of the relative likelihood: $\delta(ABC) := \max(\frac{\alpha(ABC)}{\alpha(AB)\cdot\alpha(C)}, \frac{\alpha(ABC)}{\alpha(A)\cdot\alpha(BC)})$, and pruning those below a threshold.

Algorithm 2 ConceptPhraseIdentification

Input: A document s; Number of desired concept phrases n.

Output: A set $C(s)$ of n concept phrases occurring in document s.

1: Determine significant *k-grams* occurring in document s, along with their associated significance scores.
2: Retain phrases that are terminological noun phrases.
3: Retain phrases that map to Wikipedia article titles. Form a directed graph H induced by Wikipedia linkage structure over these phrases, and propagate the scores through H in r iterations to obtain smoothed scores.
4: Return a set $C(s)$ of (up to) top n phrases based on their smoothed scores.

4.2 Identification of Core Concept Phrases

In our model, the set of core concept phrases associated with a document is intended to succinctly capture the focus or the essence of the document. Hence, it is desirable for this set to balance between the following two desirable properties. On the one hand, we would like each core concept phrase to be similar or related to as many phrases in the document as possible. On the other hand, we would like two core concept phrases to be about different aspects of the focus of the document, and hence be similar to different collection of concept phrases in the document. Hence, we define the set of core concept phrases $\Gamma(s)$ associated with document s to be the set of γ concept phrases in U that "cover" maximum number of concept phrases occurring in document s. We formalize the notion of "cover" in terms of the following optimization problem:

MaxDocumentFocus

$$\max_{\Gamma(s)\subseteq U, |\Gamma(s)|\leq\gamma} \left| \left(\bigcup_{\tilde{c}\in\Gamma(s)} B(\tilde{c}, r) \right) \bigcap C(s) \right|,$$

where $B(c, r)$ denotes the ball of radius r around the concept phrase c, that is, the set of concept phrases that are reachable within r hops from c in $\mathcal{G}$. In other words, the goal is to select a set of up to γ concept phrases in U that maximize the number of concept phrases in s that can be covered by forming balls of a small constant radius around these γ phrases.

While this objective function considers all possible sets of size at most γ in U and grows balls around the phrases in these sets, we make use of the fact that c' is part of $B(c, r)$ if and only if c is part of $B(c', r)$. Algorithm 3 provides an efficient approximation to this objective function by first greedily choosing the phrase in U that covers the maximum number of concept phrases in $C(s)$, and then iteratively choosing the phrase in U that covers the maximum number of uncovered concept phrases in $C(s)$ until the desired number γ of core concept phrases are obtained. The indicator function $1_X(x)$ evaluates to 1 if $x \in X$ and 0 otherwise.

Claim 4.1. MaxDocumentFocus *is NP-hard. Algorithm* CoreConceptPhraseIdentification *achieves an approximation ratio of* $(1 - 1/e)$ *for* MaxDocumentFocus.

Algorithm 3 CoreConceptPhraseIdentification

Input: A document s; Set $C(s)$ of concept phrases occurring in document s; Universal concept graph $\mathcal{G}$; Number of desired core concept phrases γ; Number of hops for relationship neighborhood r.
Output: A set $\Gamma(s)$ of γ core concept phrases associated with document s.

1: $\Gamma(s) := \phi$.
2: $C_{rem} := C(s)$.
3: $C_{cand} := \bigcup_{c \in C(s)} B(c, r)$.
4: **while** $(|\Gamma(s)| < \gamma)$ **and** $(|C_{rem}| > 0)$ **do**
5: $\quad \tilde{c}_{cur} := \arg\max_{\tilde{c} \in C_{cand}} \sum_{c \in C_{rem}} 1_{B(c,r)}(\tilde{c})$.
6: $\quad \Gamma(s) := \Gamma(s) \cup \{\tilde{c}_{cur}\}$.
7: $\quad C_{rem} := C_{rem} \setminus B(\tilde{c}_{cur}, r)$.
8: Return $\Gamma(s)$.

Proof. The NP-hardness proof proceeds via an efficient reduction from the Maximum Coverage problem [18], which has been shown to be NP-hard. Given an instance of Maximum Coverage problem, construct an instance of MaxDocumentFocus as follows. Set $C(s)$ to be same as the universal set X. Corresponding to each subset $S_i \subseteq X$, we create a phrase s_i and add edges to the elements of S_i. We can observe that the Maximum Coverage objective is the same as MaxDocumentFocus objective in the constructed instance, wherein we consider balls of unit radius ($r = 1$).

The proof of the approximation guarantee follows by observing that Algorithm 3 is equivalent to the greedy algorithm for Maximum Coverage problem, which has been shown to achieve an approximation ratio of $(1-1/e)$. Given an instance of MaxDocumentFocus, construct an instance of Maximum Coverage problem as follows. Define the universal set X of elements to be $C(s)$. The collection of subsets of X, $\mathcal{S} = \{S_1, S_2, \ldots, S_l\}$ is obtained as follows. First, consider the set $C_{cand} = \bigcup_{c \in C(s)} B(c, r)$, and denote its elements as $c_1, c_2, \ldots, c_l$. For $1 \leq i \leq l = |C_{cand}|$, define S_i to be $B(c_i, r) \cap C(s)$. Then, our objective function is equivalent to selecting a subset $\mathcal{S}' \subseteq \mathcal{S}$ consisting of at most γ sets such that the number of covered elements $\left|\bigcup_{S_i \in \mathcal{S}'} S_i\right|$ is maximized. Having established this reduction, we can observe that Algorithm 3 is equivalent to greedily picking the best set in each iteration until γ sets are picked, resulting in the approximation factor of $(1 - 1/e)$. □

4.3 Query Formation and Aggregation

Search engine probes in a lower dimensional space: We form queries by combining each core concept phrase with concept phrases in its relationship neighborhood. In this manner, a query combines a phrase capturing the overall focus of the document with a phrase providing additional detail. Formally, we combine each core concept phrase $\tilde{c}$ in $\Gamma(s)$ with each concept phrase in $B(\tilde{c}, r) \cap \Pi(s)$, that is, with each concept phrase present in the document representation that is also present in the ball of a small radius around the core concept phrase. Thus, we issue a total of $\Delta = \sum_{\tilde{c} \in \Gamma(s)} |B(\tilde{c}, r) \cap \Pi(s)|$ queries to the search engine. Depending on the desired media type (*e.g.*, text, image, or video), we use a search engine SE that returns relevant search results from the corresponding vertical.

Computation of most similar web documents: We obtain the most similar web documents by aggregating over the result lists across all queries for document s. The underlying intuition is that a web document occurring as a top result for multiple queries has greater similarity and relevance to the given document than a web document that occurs as a top result for only one query. The relevance score, $\nu_s(d)$ for a web document d is obtained as the sum of a position discounted score over the result lists in which d is present. In the absence of the position discounting, the relevance score equals the number of result lists in which d is present. Our algorithm returns top m web results based on this score.

We next provide formal guarantees for our algorithm. For the purposes of our analysis, we consider an ideal search engine that can retrieve all relevant web documents of a desired media type for a given (short) query, pruning irrelevant or spam documents. We assume that given a query formed by combining concept phrases, the search engine returns any candidate web document d whose representation $\Pi(d)$ contains every concept phrase in the query. This assumption, though not practical, helps us to understand the theoretical characteristics of our algorithm. In §5, we empirically demonstrate the efficacy of our algorithm using real search engines.

Claim 4.2. *Algorithm* CGSimilarity *returns web documents most similar to the given document s if:*

1. *the similarity measure, $\text{SIM}_s(d)$ is defined such that $sim_{\mathcal{G},\tilde{c}}(X, Y) := \beta_{\tilde{c}}(G_{B(\tilde{c},r)\cap X}, G_{B(\tilde{c},r)\cap Y})$, and $\beta_{\tilde{c}}(H_1, H_2)$ is chosen to be the number of common edges containing $\tilde{c}$ in both graphs H_1 and H_2, and*
2. *all relevant results are obtained for each query and the relevance score, $\nu_s(d)$ for web document d is computed as the number of result lists in which d is present.*

Proof. The proof proceeds by showing that $\nu_s(d)$ equals $\text{SIM}_s(d)$. Recall our assumption that a web document d will be part of the result list for a query formed by combining concept phrases c_1 and c_2 if and only if $c_1, c_2 \in \Pi(d)$. Consider a core concept phrase $\tilde{c} \in \Gamma(s)$ that is also present in $\Pi(d)$. Out of a total of $|B(\tilde{c}, r) \cap \Pi(s)|$ queries associated with $\tilde{c}$, document d will be present in the result lists for $|B(\tilde{c}, r) \cap \Pi(s) \cap \Pi(d)|$ queries, or equivalently, the number of common edges containing $\tilde{c}$ in both $G_{B(\tilde{c},r)\cap\Pi(s)}$ and $G_{B(\tilde{c},r)\cap\Pi(d)}$. Aggregating over all core concept phrases, we get $\nu_s(d) = \text{SIM}_s(d)$. □

5. EXPERIMENTS

In this section, we evaluate the various components of our model. To make the evaluation concrete, we base the evaluation on an application that we described in §1, in particular, augmenting electronic versions of printed textbook sections with relevant educational videos from the web.

5.1 Dataset

In tune with our application, the dataset used in our experiments consists of a diverse corpus of high school and graduate level textbooks covering Economics, Science and Math (published by the National Council of Educational Research and Training, India), and Genetics and Molecular Biology [29]. We randomly chose sections from multiple chapters in each book, obtaining a total of 70 sections. We

treated each such section as a document. The goal is to evaluate the performance of CGSimilarity in augmenting these documents with videos.

5.2 User Studies

We conducted two user studies using Amazon Mechanical Turk (AMT) platform to generate ground truth sets for evaluating various components of the model.

5.2.1 Concept Phrases

The first user study is used to construct the ground truth of concept phrases for all the documents in the dataset.

Setup: Each Human Intelligence Task (HIT) consists of the following: A judge is asked to read a document and identify top five phrases that best describe that document. Each HIT was judged by 25 judges.

Reliability of judgments: To validate the judgments, we used Fleiss Kappa measure of inter-annotator agreement. Note that this measure is designed for judges choosing from a small, common set of labels, whereas our judges were free to select arbitrary phrases. Further, since each judge identifies only five phrases, the judges may choose to skip synonymous or closely related phrases in favor of phrases that capture other topics described in the document. Therefore, we first formed the common set of labels by taking union of phrases across all judges. Then, for each judge, we assigned a score of 1 to each phrase explicitly identified by the judge, and imputed a score between 0 and 1 for each of the remaining phrases based on its similarity with the set of phrases identified by the judge (§3.2). The Fleiss Kappa measure of agreement averaged across the documents was around 0.25 indicating a fair agreement. We also analyzed the reason for fair agreement: We found that in documents that discuss multiple topics, judges selectively chose a very small subset of these topics (and hence phrases that capture only that subset). Hence, the overall agreement gets degraded, though the agreement of phrases within a 'topic' tends to be high.

Dataset A: For each section s, let $C^G(s)$ denote the ground truth set of concept phrases, that is, the union of all the phrases suggested by the judges. This dataset is used in §5.3 to evaluate ConceptPhraseIdentification.

5.2.2 Query Formation

For each document, we also constructed a ground truth set of queries that users are likely to pose in order to obtain web content relevant to the document.

Setup: Each HIT consisted of the document and the set of concept phrases, $C^A(s)$ identified using ConceptPhraseIdentification. The concept phrases were presented in alphabetical order to avoid presentation bias. The judges were asked to read the document and form at most three queries that they are likely to issue to a search engine, if they would like to obtain web content (such as videos) that are pertinent to the corresponding document. They were asked to form each query by composing two concept phrases from the provided set. We obtained 30 judgments for each HIT, and the judges were required to spend at least 30 minutes on the HIT. HITs failing this criterion were abandoned.

We can view each HIT as a session in which a user reading the text searches for relevant web content by posing one or more queries that they identified. In particular, let $Q^{j,U}(s)$ be the set of queries that judge j identified for document s. For a query $q \in Q^{j,U}(s)$, let $q.x$ and $q.y$ denote the concept phrases that make up the query. Denote the set of queries across all judges for a section s by $Q^U(s) = \cup_j Q^{j,U}(s)$. We also maintain an ordered set of concept phrases used in the queries, $C^U(s) = \cup_{q \in Q^U(s)} \{q.x, q.y\}$, where the order is defined by the usage frequency across judges.

From this user study, we constructed the following three datasets for subsequent evaluations.

Dataset B: Ordered set of concept phrases $C^U(s)$ such that $\bar{r}^s_c$ provides the ranking of $c \in C^U(s)$.

Dataset C: The set of queries for each document and judge pair, $Q^{j,U}(s)$

Dataset D: The set of queries for each document, $Q^U(s)$. For each document s, we also associate an undirected graph $\mathcal{G}^{U,s}$ defined over the set of concept phrases, $C^U(s)$. An edge exists between two concept phrases that make up a query in $Q^U(s)$.

$C^G(s)$	$C^A(s)$
electromagnetic radiation	electromagnetic waves
electromagnetic waves	radio waves
ground wave	mobile radio
frequency bands	base station
ground wave emergency network	em waves
umts frequency bands	surface wave
radio waves	base transceiver station
satellite communication	intercom
surface wave propagation	mel frequency bands
surface wave	station to mobile

Table 1: Table shows the top 10 concept phrases in $C^G(s)$ (*Dataset A*) and $C^A(s)$ for document s on Electromagnetism from Science textbook.

5.3 Identification of Concept Phrases

The goal of this experiment is to evaluate the quality of the set of concept phrases, $C^A(s)$ identified using ConceptPhraseIdentification (Algorithm 2). We evaluate the performance with respect to ground truth set described in *Dataset A* (§5.2.1).

A straight forward approach for evaluation is to compute precision and recall by measuring exact overlap between the concept phrases in $C^G(s)$ and $C^A(s)$. However, such a computation can underestimate precision and recall because it does not take into account semantics and relatedness of concept phrases. To motivate this further, consider Table 1. While there are many phrases that are not shared between the two sets, they are semantically related to each other. For instance, the phrase 'ground wave' in the context of radio transmission refers to the surface wave that propagates close to the surface of the Earth.

Metrics: In order to capture this semantic relatedness, inspired by weighted precision for ratings [28], we devise a modified metric of precision and recall that takes into account relatedness between any two concept phrases. We can compute the relatedness of a phrase $c \in C^A(s)$, to the set $C^G(s)$ by considering similarity of c with all $c' \in C^G(s)$

(§3.2). We define weighted precision and recall as follows:

$$\text{Precision}(@\theta) = \frac{\sum_{c_1 \in C^A(s)} I[\left(\sum_{c_2 \in C^G(s)} sim(c_1, c_2)\right) \geq \theta]}{|C^A(s)|} \quad (1)$$

and

$$\text{Recall}(@\theta) = \frac{\sum_{c_1 \in C^G(s)} I[\left(\sum_{c_2 \in C^A(s)} sim(c_1, c_2)\right) \geq \theta]}{|C^G(s)|}, \quad (2)$$

where $sim(c_1, c_2) = \frac{|B(c_1,1) \cap B(c_2,1)|}{|B(c_1,1) \cup B(c_2,1)|}$. Note that both precision and recall lie in the range $[0, 1]$ with higher values corresponding to higher precision and recall, respectively. In our experiments, we did not see significant difference in performance for θ up to 0.3, after which we saw a drop in recall, but no change to precision.

Results: Figure 1 shows the scatter plot between precision and recall, with each point corresponding to a document. We find that a majority of the documents have both high precision and recall. We also analyzed the documents that have lower recall in comparison. For these documents, the number of concept phrases that our algorithm found was much smaller than in ground truth, leading to reduced recall. When we studied the documents with lower values of precision, we found that this mostly happens when our algorithm identified concept phrases with non-zero similarity with phrases in $C^G(s)$ but did not meet the minimum threshold θ needed to be labeled as related.

Comparison with tf-idf: Figure 1 also shows the comparison with tf-idf. We identify top 50 tf-idf phrases to use as the representation for each section. The phrases corresponded to unigrams, and noun phrases (obtained using a parser) identified in the section. We can observe that both precision and recall values for all the sections are much lower than the proposed algorithm. We investigated the reason: Unlike a corpus of web documents, each section in a long document has a unique focus [7], and hence tends to contain distinctive phrases that are often used in that limited context (often, in just that section). These phrases have high idf. However, these phrases also have low tf since each section has a reasonable size and does not have significant redundancy. This combination results in these distinctive phrases to have lower tf-idf than a phrase that is found in several sections. However, the ground truth would have identified such distinctive phrases. On the contrary, our concept identification algorithm takes into account the relationship between phrases, and uses that information to guide the ranking of the phrases. Hence, we identify phrases that are closer to the ground truth resulting in higher precision and recall.

5.4 Ranking of Concept Phrases

In this experiment, we would like to quantify the goodness of the concept phrases ordering as obtained from Algorithm 2. Since the end task is to use combinations of these concept phrases to formulate queries for retrieval, we like to measure goodness with respect to an ordering that would be induced based on the usage of the concept phrases in the queries. To this effect, we perform comparative analysis with *Dataset B* (§5.2.2).

Metric: For any phrase c, let r_c^s and $\bar{r}_c^s$ denote the rank of c in $C^A(s)$ and $C^U(s)$ (*Dataset B*), respectively. Then, we measure goodness as the difference in the two rankings,

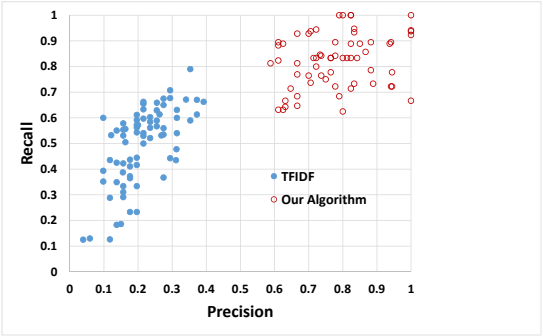

Figure 1: Precision and Recall of the concept phrases identified.

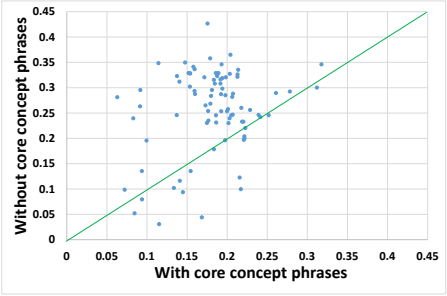

Figure 2: Effect of core concept phrases on the distance between the orderings of the concept phrases between $C^A(s)$ and $C^U(s)$.

provided by the sum of absolute differences in the inverse of ranks for each concept phrase [13]:

$$R(C^A(s)) = \sum_{c \in C^A(s)} \left| \frac{1}{r_c^s} - \frac{1}{\bar{r}_c^s} \right|. \quad (3)$$

Note that this measure penalizes more heavily when the higher ranked phrases disagree in their ranking. Thus, it inherently captures the importance of the ranks at which the phrases are positioned. Smaller the value of $R(C^A(s))$, the larger is the goodness of the ordering.

In order to understand the importance of core concept phrases, we also considered goodness for two variants of input: ordering with and without the inclusion of core concept phrases. In addition to the concept phrases, we used CoreConceptPhraseIdentification to obtain at most $\gamma = 10$ core concept phrases with relationship neighborhood of one hop ($r = 1$).

Results: Figure 2 shows the scatter plot comparing the two variants. Each point in the plot corresponds to a document. Generally, we find the distances to be small indicating that our algorithm infers an ordering over the concept phrases that best captures their relative importance to the document under consideration. In addition, when core phrases are considered, the distance is substantially smaller (mean 0.18 with standard deviation 0.05 *vs.* mean 0.26 with standard deviation 0.08). This trend is seen across most documents, as depicted by a large fraction of points lying above the 45° line.

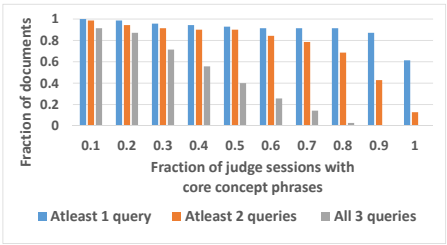

Figure 3: Usage of core concept phrases by the judges in forming queries.

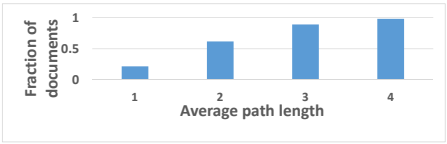

Figure 4: Query reachability: Shows the fraction of documents for which CGSimilarity found queries that are on average reachable within the reachability distance shown in the X-axis.

5.5 Query Formation: Importance of Core Concept Phrases

Next, we would like to investigate whether the core concept phrases identified by CoreConceptPhraseIdentification (§4.2) are, indeed, important for the document under consideration. If they are, then users would use these phrases in the queries they construct to retrieve web objects.

To verify this hypothesis, we performed a study using *Dataset C*. In particular, for each document, we computed the fraction of judges who formulated at least x ($x = 1, 2, 3$) queries that make use of at least one core concept phrase. Figure 3 shows the results of this evaluation. The x-axis shows the fraction of judge sessions in which at least x queries had a core concept phrase. The y-axis shows the fraction of documents in which this was true. We can see from the figure that 90% of judges pose at least one query with core concept phrase in their session, for 90% of the documents. We get similar, albeit slightly reduced, performance when we increase to $x = 2$. The fall-off is more drastic for $x = 3$. This is expected for two main reasons: First, even when there are multiple core concept phrases (corresponding to different topics), within a session, a judge typically would focus on a single topic for which she would like to get additional content. Hence, she may choose to use other non-core phrases to cover the entire spectrum of the topic of interest. Second, we also found documents, for instance in genetics book, that are tightly focused with a very few (around 2 to 3) core concept phrases, but a number of supportive non-core concept phrases which the judges used extensively. For instance, for the document on prokaryotic cell structure, the only core phrase is 'prokaryote' and all non-core phrases are terms pertaining to cell structure such as 'prokaryotic cell wall'. For this document, we found that overwhelmingly, judges used the core concept phrase in only one of their queries.

5.6 Precision of Query Formation

Here, we evaluate the performance of our algorithm in forming queries (§4.3). Let $Q^A(s)$ be the set of queries corresponding to section s that our technique identified for probing the search engine. Each $q \in Q^A(s)$ is composed of two concept phrases $q.x$ and $q.y$. The goal is to measure how well $Q^A(s)$ captures $Q^U(s)$ from *Dataset D* (§5.2.2).

One can evaluate the performance by directly computing query overlap between $Q^A(s)$ and $Q^U(s)$. However, such a computation can widely underestimate the true overlap for the following reason: Given a set of concept phrases, there are many ways in which queries can be constructed to obtain relevant web content. Therefore, we propose measuring the effectiveness of $Q^A(s)$ in capturing the overall semantics captured in $Q^U(s)$, operationalized using the notion of reachability. Given the graph $\mathcal{G}^{U,s}$ constructed using $Q^U(s)$, we measure the extent to which the concept phrases that make up the queries in $Q^A(s)$ are reachable from each other.

Metric: We measure the precision of query overlap by averaging the reachability of every query in $Q^A(s)$ with respect to the graph $\mathcal{G}^{U,s}$:

$$\text{reachable}_s = \frac{\sum_{q \in Q^A(s)} dist(q, \mathcal{G}^{U,s})}{|Q^A(s)|}, \tag{4}$$

where $dist(q, \mathcal{G}^{U,s})$ is the path length between the underlying concept phrases, $q.x$ and $q.y$ in $\mathcal{G}^{U,s}$. When no path exists between $q.x$ and $q.y$, we set this value to be large at 10. When the queries in $Q^A(s)$ are exactly found in $Q^U(s)$, then every path length is exactly one and hence $\text{reachable}_s = 1$.

Results: Figure 4 shows the results. In 20% of the documents, $Q^A(s)$ overlaps exactly with $Q^U(s)$. By allowing one extra edge (path length = 2), we substantially increase the fraction of documents fully overlapping from 20% to 60%. With two additional edges (path length = 3), we cover almost all sections. Thus, the queries formed by our algorithm are reasonably consistent with those formed by the judges.

We also studied the improvement in query overlap, if we allowed only one extra edge, in between. Figure 5 depicts this improvement. We can see that with such an extra edge, for a large number of documents, the fraction of queries that overlap with the ground truth is drastically improved. We also studied the documents in which we did not see any improvement. Often, queries in these documents had a concept phrase that is more general that required multiple edges to be reachable. As an example, consider a document from genetics book that discusses 'viral matrix protein'. This corresponds to the point (0.5, 0.5) in Figure 5. The queries that our algorithm identified for this document are provided in Table 2. The concept phrase 'enamel matrix derivative' is distantly related to 'matrix proteins' and 'viral matrix protein' that they are not reachable with an extra edge. By increasing the number of edges to four, the two concept phrases are reachable. Hence, for this document, we did not see improvement in overlap when we considered an extra edge. However, these queries together capture the underlying semantics of the document.

5.7 Finding Relevant Web Content

In this experiment, we investigate the effectiveness of CGSimilarity in identifying web content relevant to a document. For evaluation purposes, we restrict web content to be videos.

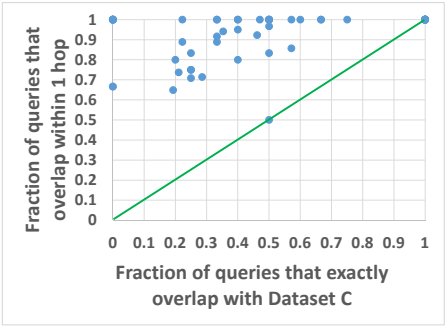

Figure 5: For each document, the fraction of queries in $Q^A(s)$ that exactly overlap with $Q^U(s)$ in *Dataset D* is shown against the corresponding fraction if we allow for an extra hop according to $\mathcal{G}^s(V, E)$).

enamel matrix derivative	matrix proteins
enamel matrix derivative	viral matrix protein
extracellular matrix	matrix proteins
extracellular matrix	viral matrix protein
viral matrix protein	matrix proteins

Table 2: Queries formed by our algorithm for a document on 'viral matrix protein'. Each row corresponds to a query and the two columns correspond to the concept phrases that make up the corresponding query.

We obtained similar results for images and text articles as well (details omitted due to space constraints).

Unlike the previous experiments, constructing ground truth set for this experiment is impractical. There are multiple videos available for the same content that it is impossible to exhaustively identify all the videos relevant to a document. In addition, with a web search engine, it is not possible to constrain the search over a smaller curated corpus. Therefore, we do comparative performance analysis.

Methods for comparison

We compare with three methods, both based on the availability of identified concept phrases corresponding to a document:

Random: A reasonably obvious approach is to construct multiple queries by randomly pairing concept phrases and aggregating their results to obtain possibly relevant videos. We use the same number of queries as CGSimilarity. We denote this technique as *Random*, and use it for comparison so as to showcase the importance of selectively choosing queries for aggregation.

Manual: We also would like to quantify the efficacy of our approach in mimicking the videos that a large fraction of the users would have desired. For this, we issue all the queries in $C^U(s)$ in *Dataset C* to retrieve videos and aggregate their results to obtain relevant videos to the document. This method will serve in lieu of constructing ground truth set of relevant videos. The goal for including this algorithm is to evaluate how close we can get to its performance.

Comity : We also compare with Comity algorithm [1] in order to compare the efficacy of our approach in effectively

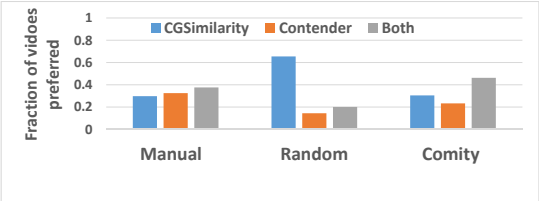

Figure 6: Pairwise comparison of CGSimilarity with the three contender methods for identifying relevant videos for documents.

retrieving videos with a much smaller number of queries. In particular, a faithful implementation of Comity requires issuing $\binom{n}{2} + \binom{n}{3}$ queries, where n is the number of concept phrases. In contrast, our proposed approach identifies a small subset of queries that is sufficient for retrieval. Following [1], we set $n = 20$ resulting in 1330 queries per document.

For all four methods, *Random*, *Manual*, *Comity*, and CGSimilarity, we obtained $t = 50$ results per query and then aggregated to obtain top $m = 3$.

Experimental setup

We conducted pairwise evaluation using Amazon Mechanical Turk. Each (HIT) consists of a document and a pair of equally ranked videos from two algorithms: one from CGSimilarity and the other from the contender (*Manual*, *Random*, or *Comity*). For each pair, we randomly permuted the order for each HIT. Each pair was judged by 5 judges. Each judge was asked to read the document, watch the two videos and specify which of the two was more relevant (or choose "Both are comparable"). They were required to spend at least 30 minutes on the task.

Results

Figure 6 shows the main results. For each contender (*Manual* or *Random*), we show three bars corresponding to the fraction of videos in which (a) CGSimilarity was preferred, (b) contender was preferred, and (c) both were equally preferred. Our approach performs comparably with *Manual* indicating that the proposed techniques identified videos that would have been preferred by a large number of judges who identified the queries pertaining to that section for retrieval of web content (*Dataset C*). We also investigated the cases in which *Manual* was preferred over our technique. Many of these cases corresponded to videos that captured aspects of the text that are mentioned in the document, but are not central to it. As a concrete example, for the document in the economics book on foreign investment and its effect on tax and non-tax revenues, the judges preferred the video on a contemporary news in India about protest against the price rise and the decision to privatize the profit making public sector undertakings. For this section, CGSimilarity picked a video on HPCL, a fortune 500 company and the largest public sector undertaking in India that was halted from being privatized.

From the same figure, we can also see that our approach performs significantly better than *Random*. In fact, *Random* exhibits huge variance in performance as shown in Figure 7. This figure shows five random trials, with each trial corre-

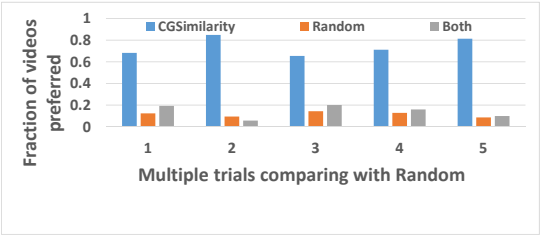

Figure 7: Comparative performance between CGSimilarity **and** *Random* **for multiple trials of** *Random*. **While both techniques formed queries using the same set of concept phrases, the performance of** *Random* **where we randomly choose the subset of queries is quite unpredictable, and much less preferred than** CGSimilarity.

sponding to a different instantiation of *Random*. While our proposed method is preferred significantly over *Random*, the figure also exhibits the variance in performance of the latter. Thus, it is imperative to selectively form the queries that are used in aggregating results from web search in order to effectively retrieve relevant content.

We can also see from the figure that CGSimilarity performs on par or better than *Comity* even though it uses much fewer number of queries (CGSimilarity used 37 queries on average, with a standard deviation of 34, compared to 1330 queries in *Comity*). Given the variance in the number of queries in a document, we also investigated if there is any correlation between the number of queries identified and the performance of our approach. We found this correlation to be quite low showing that our approach is quite resilient even when a small set of queries is issued for retrieval.

6. CONCLUSIONS

Motivated by emerging applications such as e-readers that warrant a different approach from traditional search applications, we proposed a novel theoretical model and efficient retrieval system for retrieving relevant web content of a desired media type for a given document, that only requires an oracle access to a traditional search engine. Through extensive experiments over a corpus of textbooks, we demonstrated the efficacy of our system for augmenting text documents with high quality videos from the web.

7. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Kannan, and K. Kenthapadi. Enriching textbooks with images. In *CIKM*, 2011.

[2] S. Amini, A. J. B. Brush, J. Krumm, J. Teevan, and A. K. Karlson. Trajectory-aware mobile search. In *CHI*, 2012.

[3] J. Bai, J.-Y. Nie, G. Cao, and H. Bouchard. Using query contexts in information retrieval. In *SIGIR*, 2007.

[4] I. Bordino, C. Castillo, D. Donato, and A. Gionis. Query similarity by projecting the query-flow graph. In *SIGIR*, 2010.

[5] P. D. Bruza, D. W. Song, and K.-F. Wong. Aboutness from a commonsense perspective. *Journal of the American Society for Information Science*, 51(12), 2000.

[6] H. Cao, D. Jiang, J. Pei, E. Chen, and H. Li. Towards context-aware search by learning a very large variable length hidden Markov model from search logs. In *WWW*, 2009.

[7] M. Chambliss and R. Calfee. *Textbooks for Learning: Nurturing Children's Minds*. Wiley-Blackwell, 1998.

[8] W. Dong, M. Charikar, and K. Li. Asymmetric distance estimation with sketches for similarity search in high-dimensional spaces. In *SIGIR*, 2008.

[9] W. Dong, M. Charikar, and K. Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *WWW*, 2011.

[10] C. Eickhoff, K. Collins-Thompson, P. N. Bennett, and S. T. Dumais. Personalizing atypical web search sessions. In *WSDM*, 2013.

[11] J. Etzold, A. Brousseau, P. Grimm, and T. Steiner. Context-aware querying for multimodal search engines. In *MMM*, 2012.

[12] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *IJCAI*, 2007.

[13] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW*, 2009.

[14] D. Guan, S. Zhang, and H. Yang. Utilizing query change for session search. In *SIGIR*, 2013.

[15] A. Hassan. Identifying web search query reformulation using concept based matching. In *EMNLP*, 2013.

[16] B. Hidasi and D. Tikk. Context-aware recommendations from implicit data via scalable tensor factorization. *CoRR*, abs/1309.7611, 2013.

[17] B. Hjørland. Towards a theory of aboutness, subject, topicality, theme, domain, field, content ... and relevance. *Journal of the American Society for Information Science and Technology*, 52(9), 2001.

[18] D. S. Hochbaum. Approximation algorithms for the set covering and vertex cover problems. *SIAM Journal on Computing*, 11(3), 1982.

[19] W. J. Hutchins. On the problem of aboutness in document analysis. *Journal of Informatics*, 1(1), 1977.

[20] D. Jiang, K. W.-T. Leung, and W. Ng. Context-aware search personalization with concept preference. In *CIKM*, 2011.

[21] D. Jurafsky and J. Martin. *Speech and language processing*. Prentice Hall, 2008.

[22] J. S. Justeson and S. M. Katz. Technical terminology: Some linguistic properties and an algorithm for indentification in text. *Natural Language Engineering*, 1(1), 1995.

[23] C. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[24] O. Medelyan, D. Milne, C. Legg, and I. Witten. Mining meaning from Wikipedia. *International Journal of Human-Computer Studies*, 67(9), 2009.

[25] D. F. Nettleton, L. Calderón-Benavides, and R. A. Baeza-Yates. Analysis of web search engine query session and clicked documents. In *WEBKDD*, 2006.

[26] D. Paranjpe. Learning document aboutness from implicit user feedback and document structure. In *CIKM*, 2009.

[27] S. B. Roy and K. Chakrabarti. Location-aware type ahead search on spatial databases: Semantics and efficiency. In *SIGMOD*, 2011.

[28] T. Sakai. On the reliability of information retrieval metrics based on graded relevance. *Information Processing & Management*, 43(2), 2007.

[29] R. Schleif. *Genetics and Molecular Biology*. Johns Hopkins University Press, 1993.

[30] M. Strube and S. Ponzetto. WikiRelate! Computing semantic relatedness using Wikipedia. In *AAAI*, 2006.

[31] B. Tan and F. Peng. Unsupervised query segmentation using generative language models and Wikipedia. In *WWW*, 2008.

[32] A. Ukkonen, C. Castillo, D. Donato, and A. Gionis. Searching the Wikipedia with contextual information. In *CIKM*, 2008.

[33] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li. Context-aware ranking in web search. In *SIGIR*, 2010.

[34] X. Xue, S. Huston, and W. B. Croft. Improving verbose queries using subset distribution. In *CIKM*, 2010.

[35] X. Yan, P. S. Yu, and J. Han. Substructure similarity search in graph databases. In *SIGMOD*, 2005.

[36] Y. Yang, N. Bansal, W. Dakka, P. Ipeirotis, N. Koudas, and D. Papadias. Query by document. In *WSDM*, 2009.

[37] Z. Zeng, A. K. H. Tung, J. Wang, J. Feng, and L. Zhou. Comparing stars: On approximating graph edit distance. In *VLDB*, 2009.

[38] J. Zhong, H. Zhu, J. Li, and Y. Yu. Conceptual graph matching for semantic search. In *ICCS*, 2002.

[39] R. Zhong, J. Fan, G. Li, K.-L. Tan, and L. Zhou. Location-aware instant search. In *CIKM*, 2012.