On Content-Driven Search-Keyword Suggesters for Literature Digital Libraries

Sulieman Bani-Ahmad and Gultekin Ozsoyoglu
EECS Dept. Case Western Reserve University, Cleveland OH 44120
Online Literature Digital Libraries
Current Status

- No Search keyword-suggesters
  - IEEE Explore
  - ACM Digital Library
  - PubMed
  - ScienceDirect

- Search History-Based Keyword-Suggester
  - Google Scholar – through Google Suggest

- Content-Driven Keyword-Suggester
  - The CompleteSearch engine, experimental
Why Search-Keyword Suggestion?

- helpful for constructing search keywords for queries that are
  - more accurate,
  - less ambiguous, and
  - more focused
- users spend considerable amounts of time in search sessions
  - to properly select keywords, and
  - to modify their search keywords in order to successfully locate relevant documents.
Google Suggest Approach

**Left:** No suggestions, however, relevant documents exist.  
**Up:** Suggesting from Search history and the expected output sizes.
Search-History Based SKS

- **Google's SK-Suggester**
  - utilizes the search history of all users as keyword suggestions
  - recommends keywords from
    - (i) popular searches,
    - (ii) searches from the current user’s search history,
    - (iii) current user’s bookmarks.

- **Drawbacks**
  - Noisy keywords; typos from users included as suggestions
  - Incorrect search-characterization, not well-characterized search terms.
  - Bias to users’ interests
  - Needs comprehensive search history maintenance
  - High processing and maintenance costs
The CompleteSearch Engine

Searching the CompleteSearch Engine for “query grap”

- Actual execution of query.
- Text-snippets from the document collection are returned;
  - needs preprocessing by the user.
The CompleteSearch engine Approach

- Prepare the HYB index, pre-compute inverted lists of unions of words.

- Unions of words are identified using proximity measures between words separated by $w$ (a pre-determined window size).

- Similar words are placed in the same block within the index
  - Maintains a good level of locality of search,
  - As the user enters his search words, relevant blocks determine the (searching) scope,
Content-Based SK-Suggester

- Content-Based SK-Suggester anticipates users’ search keywords.

General Approach:

- (i) parse the document collection to be searched,
- (ii) prepare offline refinements to search keywords, and
- (iii) dynamically suggest keywords as the user types his/her keywords
Our Approach

(i) Parse the document collection: *Link Grammar parser.*
   - a syntactic parser of English,

(ii) Group publications.
   - “most-specific” research topics (research pyramids),

(iii) Within each research pyramid (RP), build a hierarchical structure of simple and compound tokens (phrases), islands

(iv) Attach topic-(RP-)sensitive scores to keywords
   - use TextRank, a text summarization tool,

(v) Use the identified *research-topics* to help user choose more focused search keywords *prior to actual search query execution.*
Utilized Tools

The Link-Grammar English Parser
The Research-Pyramid Based Grouping Utility
TextRank Algorithm
Linguistic Pre-Processing Step

Tokenize documents

- transforms documents into a categorized block of text, called tokens

Form compound tokens: combine two or more simple tokens at a time

- Builds syntactically and semantically correct suggestions (using the link-grammar output)

**Example:** Tokenize the paper title: "The Linear Complexity of a Graph":

Generate simple tokens:

(i) “the”, "of" and “a” which are stopwords, and
(ii) "linear", “complexity”, and "graph" which are non-stopwords.
(iii) form compound tokens: “linear complexity”.
Link-Grammar Parser

outlier.n detection.n for.p high.a dimensional.a data.n

a model.n for.p querying.v annotated.v documents.n
### Observed linkage types and their percentages

<table>
<thead>
<tr>
<th>Linkage type</th>
<th>% across ACM SIGMOD Anthology</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>24.35</td>
<td>Connects adjective to noun</td>
</tr>
<tr>
<td>(AN)</td>
<td>23.43</td>
<td>Connects noun-modifier to noun</td>
</tr>
<tr>
<td>(J)</td>
<td>18.28</td>
<td>Connects preposition to its objects</td>
</tr>
<tr>
<td>(D)</td>
<td>8.42</td>
<td>Connects determinator to noun</td>
</tr>
<tr>
<td>(M)</td>
<td>8.69</td>
<td>Connects noun to post-noun modifiers</td>
</tr>
<tr>
<td>(MV)</td>
<td>4.46</td>
<td>Connects verbs to adjectives</td>
</tr>
<tr>
<td>(O)</td>
<td>6.15</td>
<td>Connects transitive verbs to objects</td>
</tr>
</tbody>
</table>

### Frequency of each observed parts-of-speech token

<table>
<thead>
<tr>
<th>Part of Speech</th>
<th>Frequency</th>
<th>Part of Speech</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nouns</td>
<td>47.32</td>
<td>Adverbs</td>
<td>0.076</td>
</tr>
<tr>
<td>Adjectives</td>
<td>15.23</td>
<td>Clauses</td>
<td>0.069</td>
</tr>
<tr>
<td>Verbal nouns</td>
<td>10.65</td>
<td>Relative clauses</td>
<td>0.025</td>
</tr>
<tr>
<td>Prepositions</td>
<td>4.48</td>
<td>Un-tagged</td>
<td>21.14</td>
</tr>
</tbody>
</table>
Token Hierarchy

Research Pyramids Hierarchy

Title layer

Island layer

Simple/compound

Token layer

Linkages

Token Hierarchy
The Research-Pyramid Model

- The Research Pyramid Model of Research Evolution
  
  *Publications in any scientific discipline are ‘naturally’ categorized into groups called ‘research pyramids’.

- A **research pyramid (RP)**: publications that belong to a most-specific research topic.

- Each RP has a pyramid-like structure in terms of the citation graph
The Research-Pyramid Model

- The goal of each user search session:
  - find information about a specific topic.
- (The locality of search principle):
  - **Within a single search session, the user targets documents within a specific topic (i.e., an RP).**
- SK suggestion scope
  - chosen as close to the topic being targeted as possible.
- As user enters more search keywords:
  - prune out topics (keywords) that are of low significance.
- Single token (word) significance => TextRank
TextRank Algorithm

- Used to compute Topic-Sensitive Token and Phrase Weights.
- Vertices are filtered tokens (nouns and adjectives),
  - Syntactic filter
- Links represent linguistic/textual adjacency (in our case, they appear in the same title)
- Each graph for a research-pyramid is topic-sensitive.
Advantages of Our Approach

- Eliminates the drawbacks of Google’s search history-based SK-Suggester.
- Boosts the performance of the techniques used in the CompleteSearch.
- Has an excellent locality of access.
Main Modules of the Content-Driven SK-Suggester
Procedure **SK-SuggesterInterface** ()

**Input**

User Input $w$: current search-terms

Server Input: $R$, and $I$ (stored in session status)

{

(1) For $w$

(1.1) $\text{LISK} \leftarrow$ the uncompleted search keyword in $w$

(1.2) $\text{CSK} \leftarrow$ the completed search keywords in $w$

(1.3) If ($\text{CSK} = "" \& \& \text{LISK} = ""$)

\[ \text{STA}_\text{Module}(\text{LISK}); \]

(1.4) Elseif ($\text{CSK} = "" \& \& \text{LISK} = ""$)

\[ \text{QR}_\text{Module}(\text{CSK, LISK}); \]

(1.5) Elseif ($\text{CSK} = "" \& \& \text{LISK} = ""$)

$\text{SK-List1} \leftarrow \text{STA}_\text{Module}(\text{LISK});$

$\text{SK-List2} \leftarrow \text{QR}_\text{Module}(\text{CSK, LISK});$

\[ \text{Join}(\text{SK-List1, SK-List2}) \]

(2) $\text{Presentation}_\text{Module}(W')$
Procedure SK-SuggesterInterface ()
Input
User Input w: current search-terms
Server Input: R, and I (stored in session status)
{
(1) For w
   (1.1) LISK <- the uncompleted search keyword in w
   (1.2) CSK <- the completed search keywords in w
   (1.3) If (CSK="" && LISK="")
       \textbf{STA\_Module}(LISK);
   (1.4) Elseif (CSK="" && LISK="")
       \textbf{QR\_Module}(CSK, LISK);
   (1.5) Elseif (CSK="" && LISK="")
       SK-List1 <- \textbf{STA\_Module}(LISK);
       SK-List2 <- \textbf{QR\_Module}(CSK, LISK);
       Join(SK-List1, SK-List2)
(2) \textbf{Presentation\_Module}(W')
Procedure SK-SuggesterInterface ()

Input
User Input $w$ : current search-terms
Server Input : $R$, and $I$ ( stored in session status)

{$w$}$w$}{
(1) For $w$
   (1.1) LISK <- the uncompleted search keyword in $w$
   (1.2) CSK <- the completed search keywords in $w$
   (1.3) If (CSK="" && LISK="")
      STA_Module(LISK);
   (1.4) Elseif (CSK="" && LISK="")
      QR_Module(CSK, LISK);
   (1.5) Elseif (CSK="" && LISK="")
      SK-List1 <- STA_Module(LISK);
      SK-List2 <- QR_Module(CSK, LISK);
      Join(SK-List1, SK-List2)
(2) Presentation_Module($W'$)
Procedure SK-SuggesterInterface

Input
User Input \( w \): current search-terms
Server Input : \( R \), and \( I \) (stored in session status)

\{
  (1) For \( w \)
    (1.1) LISK <- the uncompleted search keyword in \( w \)
    (1.2) CSK <- the completed search keywords in \( w \)
    (1.3) If (CSK="" && LISK="")
      \( \text{STA}_\text{Module}(\text{LISK}); \)
    (1.4) Elseif (CSK="" && LISK="")
      \( \text{QR}_\text{Module}(\text{CSK}, \text{LISK}); \)
    (1.5) Elseif (CSK="" && LISK="")
      \( \text{SK-List}1 <- \text{STA}_\text{Module}(\text{LISK}); \)
      \( \text{SK-List}2 <- \text{QR}_\text{Module}(\text{CSK}, \text{LISK}); \)
      \( \text{Join}((\text{SK-List}1), (\text{SK-List}2)); \)
  (2) \( \text{Presentation}_\text{Module}(W') \)
\}
Guiding statistics provided to the user online

- **Suggestion Scope**
  - the number of research topics (or research pyramids) where the search keywords \( w \) are observed

- **Topic-Sensitive Popularity of Search Keywords** \( \text{TSP}(W', r) \)
  - \( W' \): search terms
  - the sum of TextRank scores of all words in \( W' \)
  - Query refinements \( (W') \): presented in the order of their *matching scores*:
    - Similarity \( (W', W) \): text-based similarity between \( W' \) and the search terms \( W \)
    - \[
      M_{\text{Score}}(W', W) = \text{Similarity}(W', W) \times \max_{r \in \text{ERP}(W')} [\text{TSP}(W', r)]
    \]
Experimental Results and Observations

Accuracy of Linguistic Pre-processing and Quality of Suggestions

Scalability and Index Sizes

Convergence of Suggestion Scope
Untagged Tokens

<table>
<thead>
<tr>
<th>Token</th>
<th>% of untagged tokens</th>
<th>Token</th>
<th>% of untagged tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>of</td>
<td>17.39</td>
<td>to</td>
<td>4.72</td>
</tr>
<tr>
<td>a</td>
<td>14.76</td>
<td>an</td>
<td>4.51</td>
</tr>
<tr>
<td>in</td>
<td>14.30</td>
<td>on</td>
<td>3.43</td>
</tr>
<tr>
<td>and</td>
<td>14.12</td>
<td>with</td>
<td>3.34</td>
</tr>
<tr>
<td>the</td>
<td>11.92</td>
<td>from</td>
<td>1.62</td>
</tr>
</tbody>
</table>

The distribution of un-tagged tokens

- Un-tagged tokens are all stopwords.
- We use them
  - (i) to construct islands and
  - (ii) to connect linguistically adjacent islands.
K-word Proximity Search

- The search keywords “query graph” are already identified as one island.
- Suggesting query refinements based on islands may help towards a successful proximity search.
- Notice that item (3) is probably irrelevant to the query at search time since this publication most probably belongs to different research pyramid from the first and second hits; this false positive is pruned or pushed down in ranking query results.
- Informing users of the linguistic proximity of search terms prior to query execution can thus be useful.
- Furthermore, informing the user of the order in which terms appear may help eliminate false hits like hit (4), which is called k-word ordered proximity search.

(1) Multiple Query Processing In Deductive Databases Using Query Graphs
(2) Query Graphs Implementing Trees And Freely Reorderable Outerjoins
(3) Effective Graph Clustering For Path Queries In Digital Map Databases
(4) Query By Diagram, A Graphic Query System

Possible hits of the query “query graph”
Convergence of Suggestion Score

\[ \text{Coverage(r)} = -\log\left(\frac{\text{# of tokens used in } r}{\text{total # of tokens}}\right) \]

- Coverage values range between 3 and 8, which means that
  - (i) the tokens within each research pyramid are of low diversity, and
  - (ii) this signifies the importance of ranking tokens within research topics.
- This serves to push refinements extracted from dominant research topic(s) up in the suggestion list.

- **Observation:** Filtered tokens have limited scope.
Convergence of Suggestion Score

- Search-keyword suggestion can be optimized reducing suggestion scope
  - Suggestion scope converges fast.

- High TextRank => more significant tokens.
- Tokens that score high (>0.8): content-bearing of the corresponding research pyramid.
- Low-scored tokens: widely used tokens.
- Order SK-suggester output
  - Significant refinements from dominant research pyramids.
Future Work

- Our proposed approach can be used on any document collection given that
  - documents are clustered
  - highly topically related clusters.

- *Hybrid search-keyword suggester* (i.e. content-driven and search-history based).
  - Initially, no search-history => suggest document contents.
  - track users’ search terms
  - Build search-history.

- Why Hybrid SK-suggesters? combine the advantages of both approaches
  - (i) accurate search-history keywords, less *incomplete* or *misspelled* words,
  - (ii) well-characterized queries. Keywords from successful searches
  - (iii) personalized search-keyword suggestions. *Which is an advantage of Google’s search-history based suggester.*
Questions?

Project website:
http://dblab.case.edu:8080/giza/doku.php