Fast Generation of Result Snippets in Web Search

Felix Geller

January 20, 2009
Agenda

► What are “snippets”?  
► How do they fit in?  
► Possible difficulties  
► Speeding up snippet generation  
  ► Document compression  
  ► Document compaction  
► Summary
HPI Welcome

It was established by Professor **Hasso Plattner** who demonstrates a great deal of personal commitment to the **Institute**'s teaching and research as well as ... www.hpi.uni-potsdam.de/welcome.html?

**Hasso Plattner Institut der Uni Potsdam** - [Translate this page]

Hasso Plattner-Institut fuer Softwaresystemtechnik GmbH, Potsdam | Hasso Plattner-Institut for software systems engineering GmbH, Potsdam/Germany. www.hpi.uni-potsdam.de/ - 43k - Cached - Similar pages

More results from www.hpi.uni-potsdam.de >

Stanford **Institute of Design** | d school | home

2007 Stanford University **Institute** of Design. All Rights Reserved. Terms of Use | Copyright Complaints | Internal Login. dschool.stanford.edu/ - 20k - Cached - Similar pages

**Hasso Plattner** - Wikipedia, the free encyclopedia

16 Nov 2003 ... Also in 1998, Plattner founded the **Hasso Plattner Institute** [1] for software systems engineering based at the University of Potsdam, ...

en.wikipedia.org/wiki/Hasso_Plattner - 32k - Cached - Similar pages

**Hasso Plattner Institute** - Wikipedia, the free encyclopedia

10 Nov 2005 ... The **Hasso Plattner Institute** is a German information-technology college, affiliated to University of Potsdam and located at ...

en.wikipedia.org/wiki/Hasso_Plattner_Institute - 22k - Cached - Similar pages

More results from en.wikipedia.org >

**Potsdam de** - **Hasso Plattner Institute** for Software Systems Engineering

View of the **Hasso Plattner Institute** (HPI) in Dabelsberg ... **Address** Hasso-Plattner-Institut für Softwaresystemtechnik Prof.-Dr.-Helmert-Straße 2-3 ...

www.potsdam.de/cms/ziel/35120/EN/ - 36k - Cached - Similar pages

[PDF] **GFD-E, 103 Peter Tröger, Hasso-Plattner-Institut** Becky Gietzel ...

File Format: PDF/Adobe Acrobat - View as HTML

Hasso-Plattner-Institut, University of Potsdam. Prof.-Dr.-Helmert-Str. 2-3 ... Please address the information to the OGE Executive Director ...
Snippets are short fragments of text extracted from the document content (or its metadata). ... A query-biased snippet is one selectively extracted on the basis of its relation to the searcher’s query. [4]

→ Quickly identify relevant documents without opening the document as a whole.
Agenda

- What are “snippets”?
- **How do they fit in?**
- Possible difficulties
- Speeding up snippet generation
  - Document compression
  - Document compaction
- Summary
Abstract Search Engine Architecture

WEB

Document Storage

Term Mapping

Crawling Engine

Indexing Engine

Lexicon Engine

Ranking Engine

Snippet Engine

Meta Data Engine

Query Engine

R

Document Storage

Term Mapping

Crawling Engine

Indexing Engine

Lexicon Engine

Ranking Engine

Snippet Engine

Meta Data Engine

Query Engine

WEB
Agenda

- What are “snippets”?  
- How do they fit in?  
- **Possible difficulties** 
- Speeding up snippet generation  
  - Document compression  
  - Document compaction  
- Summary
Possible Difficulties

**Relevance**
Query-biased, i.e. non-static summary.

**Context**

**Speed**
- Storage: “order of ten billion web pages” [4]
- Load: “hundreds of millions of search queries per day” [4]
- Response: File I/O is a major bottleneck
Agenda

- What are “snippets”?
- How do they fit in?
- Possible difficulties
- **Speeding up snippet generation**
  - Document compression
  - Document compaction
- Summary
Majority of time spent generating a snippet is in locating the document on disk . . . : 64% for whole documents.

With 1% of documents cached, . . . around 80% of disk seeks are avoided.\textsuperscript{[4]}

- **Disk Cache**: Managed by OS, e.g. stores frequently accessed documents
- **Query Cache**: Stores precomputed result pages for popular queries
- **Document Cache**: Stores frequently accessed documents in main memory
Agenda

- What are “snippets”?
- How do they fit in?
- Possible difficulties
- Speeding up snippet generation
  - Document compression
  - Document compaction
- Summary
Document Compression: Concepts

- Compressed Token System (CTS)
- Document content is normalized (convert br, remove tags)
- Atomic entity: Word
- Entity of interest: Sentence
- Replace words with numbers
- vbyte coding scheme (think UTF-8)
- Words alternate non-words (i.e. punctuation)
Document Compression: Algorithm

1st Pass
- Collect words
- Collect non-words
- Construct model

2nd Pass
- Replace words and non-words
- Escape words which are not encoded
Document Compression: Example

Educators, generals, dieticians, psychologists, and parents program.

Armies, students, and some societies are programmed.

An assault on large problems employs a succession of programs,
most of which spring into existence en route.

These programs are rife with issues that appear to be particular
to the problem at hand.

To appreciate programming as an intellectual activity in its own
right you must turn to computer programming;
you must read and write computer programs -- many of them.

It doesn’t matter much what the programs are about or what
applications they serve.

What does matter is how well they perform and how smoothly they
fit with other programs in the creation of still greater programs.

Sample taken from [1]
### Word Model

<table>
<thead>
<tr>
<th>Code</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>&quot;with&quot;</td>
</tr>
<tr>
<td>1</td>
<td>&quot;you&quot;</td>
</tr>
<tr>
<td>2</td>
<td>&quot;how&quot;</td>
</tr>
<tr>
<td>3</td>
<td>&quot;are&quot;</td>
</tr>
<tr>
<td>4</td>
<td>&quot;in&quot;</td>
</tr>
<tr>
<td>5</td>
<td>&quot;computer&quot;</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Non-Word Model

<table>
<thead>
<tr>
<th>Code</th>
<th>Non-word</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>&quot; &quot;</td>
</tr>
<tr>
<td>1</td>
<td>&quot;,&quot;</td>
</tr>
<tr>
<td>2</td>
<td>&quot;,,&quot;</td>
</tr>
<tr>
<td>3</td>
<td>&quot; – &quot;</td>
</tr>
<tr>
<td>4</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>5</td>
<td>&quot;;&quot;</td>
</tr>
</tbody>
</table>
Document Compression: Example

<table>
<thead>
<tr>
<th>Educators</th>
<th>generals</th>
<th>dieticians</th>
<th>psychologists</th>
<th>parents</th>
<th>program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Armies</td>
<td>students</td>
<td>some</td>
<td>societies</td>
<td>programmed</td>
<td></td>
</tr>
<tr>
<td>An</td>
<td>assault</td>
<td>on</td>
<td>large</td>
<td>problems</td>
<td>employs</td>
</tr>
<tr>
<td>These</td>
<td>011030</td>
<td>rife</td>
<td>issues</td>
<td>that</td>
<td>appear</td>
</tr>
<tr>
<td>To</td>
<td>appreciate</td>
<td>as</td>
<td>an</td>
<td>intellectual</td>
<td>activity</td>
</tr>
<tr>
<td>10130</td>
<td>read</td>
<td>write</td>
<td>many</td>
<td>0140</td>
<td>them</td>
</tr>
<tr>
<td>It</td>
<td>doesn</td>
<td>t</td>
<td>much</td>
<td>or</td>
<td>applications</td>
</tr>
<tr>
<td>What</td>
<td>does</td>
<td>is</td>
<td>well</td>
<td>perform</td>
<td>smoothly</td>
</tr>
</tbody>
</table>

Felix Geller: Fast Generation of Result Snippets in Web Search
# Document Compression: Gain I/II

<table>
<thead>
<tr>
<th></th>
<th>WT10G</th>
<th>WT50G</th>
<th>WT100G</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. Docs ($\times 10^6$)</td>
<td>1.7</td>
<td>10.1</td>
<td>18.5</td>
</tr>
<tr>
<td>Raw Text</td>
<td>10,522 MB</td>
<td>56,684 MB</td>
<td>102,833 MB</td>
</tr>
<tr>
<td>Baseline ($zlib$)</td>
<td>24%</td>
<td>19%</td>
<td>19%</td>
</tr>
<tr>
<td>CTS (+1024 MB)</td>
<td>26%</td>
<td>21%</td>
<td>22%</td>
</tr>
</tbody>
</table>

Taken from [4]
### Document Compression: Gain II/II

<table>
<thead>
<tr>
<th></th>
<th>WT10G</th>
<th>WT50G</th>
<th>WT100G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>75</td>
<td>157</td>
<td>183</td>
</tr>
<tr>
<td>CTS</td>
<td>38</td>
<td>70</td>
<td>77</td>
</tr>
<tr>
<td>Reduction in time</td>
<td>49%</td>
<td>56%</td>
<td>58%</td>
</tr>
</tbody>
</table>

Average time (ms) for the final 7000 queries.

Taken from [4]
Agenda

▶ What are “snippets”?  
▶ How do they fit in?  
▶ Possible difficulties  
▶ Speeding up snippet generation  
  ▶ Document compression  
  ▶ Document compaction  
▶ Summary
Document Compaction: Concepts

- Reduce size of documents
  → Remove sentences which are deemed insignificant

- Reduce query time
  → Order sentences by significance
Natural order: First sentence should introduce paragraph!

Significant terms (ST): Score based on term frequency [2]

Query log based (QLt): Score based on past query terms

Query log based (QLu): Same as QLt, but considers only unique terms
Intermezzo: Sentence Ranking

Document is broken into sentences $S$ where $S = [w_1, w_2, \ldots, w_m]$. Query $Q$ where $Q = \{q_1, q_2, \ldots, q_n\}$

- $h$: sentence is a heading
- $l$: sentence is first or second line of document
- $k$: length of longest contiguous run of $q_i$ in $S$
- $c$: count of $w_j \in Q$
- $d$: $c$ minus repetitions
Impact on Sentence Ranking

Impact of omitting 50% of sentences based on ST

\[ h + l \] approximately 8% change
\[ c \] approximately 2% change
\[ k \] approximately 3% change
\[ d \] approximately 8% change

Taken from [4]
Agenda

- What are “snippets”?
- How do they fit in?
- Possible difficulties
- Speeding up snippet generation
  - Document compression
  - Document compaction
- Summary
What are “snippets”?
→ Query-biased text fragments, facilitating the identification of relevant information.

Possible difficulties
→ Relevance, Context, Speed.

Speeding up snippet generation
→ Make use of caches.
  ▶ Document compression
    → Encode words using numbers, make use of dictionary.
  ▶ Document compaction
    → Reducing document size by 50% has arguably low impact.
References


H.P. Luhn.


Anastasios Tombros and Mark Sanderson.


Andrew Turpin, Yohannes Tsegay, David Hawking, and Hugh E. Williams.


Wikipedia.

Backup Slide

Taken from [4]
Backup Slide

Taken from [4]
Backup Slide

Collection Size (Gb) or Time (msec)

Model Size (Mb)

- Size (Gb)
- Time (msec)

Taken from [4]
Backup Slide

% of accesses as cache hits

Cache size (% of collection)

- LRU Q=0
- LRU Q=10,000
- Static Q=0
- Static Q=10,000

Taken from [4]
Backup Slide

The graph shows the time (in seconds) taken for different query processing methods, grouped into 100's of queries. The legend indicates:
- Baseline
- CTS with caching
- CTS without caching

The graph suggests that CTS with caching has the lowest time, followed by Baseline, and then CTS without caching.

Taken from [4]
Let $f_{d,t}$ be the frequency of term $t$ in document $d$, then term $t$ is determined to be significant if:

$$
f_{d,t} \leq \begin{cases} 
7 - 0.1 \times (25 - s_d), & \text{if } s_d < 25 \\
7, & \text{if } 25 \leq s_d \leq 40 \\
7 + 0.1 \times (s_d - 40), & \text{otherwise}
\end{cases}
$$

where $s_d$ is the number of sentences in document $d$.

A *bracketed section* is defined as a group of terms where the leftmost and rightmost terms are significant terms, and no significant terms in the bracketed section are divided by more than four non-significant terms.

The *score for a bracketed section* is the square of the number of significant words falling in the section, divided by the total number of words in the entire sentence.

The *score for a sentence* is the maximum of all scores for the bracketed sections of the sentence.

Quoted from [4], technique based on [2], [3].