SnakeT
A personalized search engine

Paolo Ferragina
Dipartimento di Informatica, Università di Pisa
(Joint with Antonio Gulli)

Presented at WWW 2005

Goal of a Search Engine

Retrieve the docs that are “relevant” for the user query

- Doc: file word or pdf, web page, email, blog, book,...
- Query: paradigm “bag of words”
- Relevant?
  ... We face many difficulties...
Web is huge: 8 bil pages [Google]

Web is heterogeneous and dynamic

Languages/Encodings
- Hundreds of languages: 55 (Jul01)
- Home pages (1997): English 82%, Next 15: 13%
- Google (mid 2001): English: 53%, JGCFSKRIP: 30%

Distributed authorship
- Millions of people creating pages with their own style...
- Not all have the purest motives in providing high-quality information - commercial motives drive “spamming”.
Web is highly dynamic
[weekly snapshot during 2004: 154 web sites, 3-5 mil pg, 65Gb]

Figure 2: Fraction of pages from the first crawl still existing after n weeks (dark bars) and new pages (light bars).

Figure 3: Fraction of links from the first weekly snapshot still existing after n weeks (dark/bottom portion of the bars), new links from existing pages (grey/middle) and new links from new pages (white/top).

A “good” coverage of the Web is difficult, if not impossible

User Needs

- **Informational** – want to learn about something (~40%)
  - Asthma

- **Navigational** – want to go to that page (~25%)
  - Alitalia

- **Transactional** – want to do something (~35%)
  - Access a service
  - Downloads
  - Shop
  - Haifa weather
  - Mars surface images
  - Nikon CoolPix
User Queries

- ill-defined queries
  - Short
    - 2001: 2.54 terms avg
    - 80% less than 3 terms
  - Imprecise terms
  - 78% are not modified

- Wide variance in
  - Needs
  - Expectations
  - Knowledge
  - Patience: 85% look at 1 page

Evolution of Search Engines

- First generation -- use only on-page, text data
  - Word frequency and language
  - 1995-1997 AV, Excite, Lycos, etc

- Second generation -- use off-page, web-specific data
  - Link (or connectivity) analysis
  - Anchor-text (How people refer to a page)
  - From 1998. Made popular by Google, now everyone

- Third generation -- answer “the need behind the query”
  - Focus on "user need", rather than on query
  - Click-through data
  - Context determination
  - Helping the user
  - Still experimental
Current scenario  [Gulli-Signorini, www 05]

WHO’S THE BIGGEST?

Our experiments, revealed to us which engine has the greatest coverage of the Indexable Web.

ASK/TEOMA (58%)
MSN BETA (62%)
YAHOO! (69%)
GOOGLE (76%)
INDEXABLE WEB (11.5 billion pages)

Google vs Yahoo
Share 3.8 results in the top 10 on avg
Share 23% in the top 100 on avg

In summary

Current search engines incur in many difficulties:

عناوين التصنيف التوضيحي (بالإنجليزية):

- Link-based ranking may be inadequate: bags of words paradigm, ambiguous queries, polarized queries, ...
- Coverage of one search engine is poor, meta-search engines cover more but “difficult” to fuse multiple sources
- User needs are subjective and time-varying
- Users are lazy and look to few results

A new “complementary” approach is
Web-Snippet Hierarchical Clustering
Web-Snippet Hierarchical Clustering

- The **folder hierarchy** must be formed
  - "on-the-fly from the snippets": because it must adapt to query results without any costly remote access to the original web pages or documents
  - "and his folders may overlap": because a snippet may deal with multiple themes
  - **Canonical clustering is instead persistent and generated only once**

- The **folder labels** must be formed
  - "on-the-fly from the snippets" because labels must capture the **potentially unbounded themes** of the results without any costly remote access to the original pages or documents.
  - "and be intelligible sentences" because they must facilitate the user post-navigation
  - **It is NOT a "document organization into topical context" because**
    - snippets are poorly composed,
    - no "structural information" is available for them
    - "static classification" into predefined categories would be not appropriate.
The Literature

- We may identify four main approaches (i.e. taxonomy)
  - Single words and Flat clustering – Scatter/Gather, WebCat, Retriever
  - Sentences and Flat clustering – Grouper, Carrot2, Lingo, Microsoft China
  - Single words and Hierarchical clustering – FIHC, Credo
  - Sentences and Hierarchical clustering – Lexical Affinities clustering, Hierarchical Grouper, SHOC, CIIRarchies, Highlight, IBM India

- Conversely, we have many commercial proposals:
  - Northerlight (stopped 2002)
  - Copernic, Mooter, Kartoo, Groxis, Clusty, Dogpile, iBoogie,…
  - Vivisimo is surely the best!
Snake T’s main features

- 2 knowledge bases for selecting-and-ranking the labels
  - DMOZ is used as a feature selection and sentence ranker index
  - Text anchors are used for snippets enrichment

- Labels are **gapped sentences** of variable length
  - Grouper’s extension, to match sentences which are "almost the same"
  - Lexical Affinities clustering extension to k-long LAs

- Hierarchy formation deploys the folder labels and results coverage
  - Two types of labels: “primary” and “secondary” for finer/coarser clustering
  - Syntactic and covering pruning rules for simplification and compactation

- 18 engines (Web, news and books) are queried on-the-fly
  - Google, Yahoo, Teoma, A9 Amazon, Google-news, etc..
  - They are used as black-boxes

Generation of candidate labels

... **build frequent gapped sentences**

(this extends other approaches)

1) Extract all word pairs occurring in the snippets within some proximity window
2) Rank them by exploiting: KB + frequency within snippets
3) Discard the pairs whose rank is below a threshold
4) Merge remaining pairs taking into account their original position, order, and sentence boundary within the snippets

... We have so generated the *candidate labels for folder leaves*...
From candidate labels to folder leaves

*Key feature is readability* - Labels should be fairly distinct among folder descendants and siblings.

Initially build the folder leaves:

- Build one folder $C_i$ from each candidate label $L_i$. The folder is formed by all snippets containing the gapped sentence $L_i$.
  - $L_i$ is called *primary label* of the folder $C_i$.
- Enrich the folder $C_i$ with *secondary labels* $S_i$ which are gapped sentences with good rank and frequency $\geq 80\%$ within $C_i$’s snippets.
  - The *signature* of the folder $C_i$ is the string $L_i$#$S_i$#$\ldots$#$S_k$.

... We have so generated the *candidate leaves* and their labels...

---

From leaves to the whole hierarchy

- **Parent formation**: build the next level of the hierarchy
  - Find substrings shared by the folder signatures $[\Rightarrow$ *new labels*]$]
  - For each shared substring create a new parent folder.
- **Ranking**: clean-up some folders based on ranking
  - Compute the rank of the labels of these new folders
  - Discard the folders having low rank
- **Pruning**: based on hierarchy “readability” and “compactness”
  - *Context pruning*: greedy approach to a graph covering problem (NPH)
  - *Content pruning*: empirical entropy about word distribution plus ...

... We have so generated the *hierarchy of labeled folders*...
Evaluating SnakeT

Large set of experiments: 77 queries on 16 engines

- Meta-search is useful: The use of many search engines increases the number and distinctness of the extracted themes
- Text-anchors and DMOZ improve precision of the labels
- Gapped sentences improve the compaction of the hierarchy and the readability of the labels

See the paper in Proc. WWW '05
Our personalization aims for...

![Diagram showing Clustering and Ranking with SnekTE]

### Clusters
- Personaiized
- Unpersonaiized
- Unchecked

- Java
- Technology
- Programming
- Tutorials
- Free
- Training
- Developers
- Java Books
- Features Java
- Coffee
- Site For Java
- Games
- Java Index
- Java Environment
- Java Forums
- Virtual Machine

### Search
- The Java Tutorial
- Sun Microsystems Developers Home Products Technologies Java Technology Learning Tutorial Tutorial ...
  - [Google](#)
- Java(TM) Boutique - Programming Tutorials, Reviews and Downloads
- The Java Boutique is a collection of Java applets, games, samples, and tutorials. Learn programming and find demo apps about Java and Java.
  - [Google](#)
  - [altavista](#)
  - [msn](#)
  - [searcheng](#)
  - [yahoo](#)
- JavaScript Ka- Comprehensive JavaScript, DHTML, CSS tutorials and ...
- JavaScript Ka Comprehensive Website Abstracts, Click Here ...
  - [Google](#)
- Java Programming Resources -- Java, Java, and more Java
- Java programming resources: FAQs, tutorials, compiler and browser downloads, documentation, ...
  - [Google](#)
- Molecular Expressions: Science, Optics and You - Secret Worlds
- Molecular Expressions: Interactive Java Tutorial, ATTENTION... functioning properly. Please install this software ...
  - [Google](#)
- Welcome to Freewarejava.com, the place to find free Java applets...
SnakeT’s personalization

😊 Many interesting features

😊 Full-adaptivity to the variegate user needs/behaviors
😊 Scalable to many users and their unbounded profiles
😊 Privacy protection: no login or profiled information is required/used
😊 Black-box: no change in the underlying (unpersonalized) search engine

The user first gets a “glimpse” on all themes of results without scanning all of them, and then selects (filters) the results interesting for him/her.

Conclusions

☐ Check it at: snaket.di.unipi.it

☐ We are searching for:
  ■ Linguistic tools that improve the label extraction process
  ■ Re-ranking techniques that exploit the folders selected by the user during the personalization session
  ■ Deal with click-through data for better personalization and re-ranking [WWW 05]
  ■ Algorithmic approach to semantic similarity for better ranking based on DMOZ and other KBs [WWW 05]

☐ Desktop search engine