PV030 Textual Information Systems

Petr Sojka

Faculty of Informatics
Masaryk University, Brno

Spring 2012

Outline (Week ten)

- ☞ Google as a TIS example.
- ☞ Brainstorming on Google, (Watson, Alpha).
- ☞ Google – historical notes.
- ☞ Google – system architecture.
- ☞ Google – PageRank.
- ☞ Google File System.
- ☞ Implementation of index systems

Gooooooooooooogle – a bit of history

An example of anatomy of global (hyper)text information system (www.google.com).

- 1997: google.stanford.edu, students Page and Brin
- 1998: one of few quality search engines, whose basic fundamentals and architecture (or at least their principles) are known – therefore a more detailed analysis according to the article [G00]

  http://www7.conf.au/programme/fullpapers/1921com1921.htm.
- 2011: clear leader in global web search

Goooooooooooogle – anatomy

- Several innovative concepts: PageRank, storing of local compressed archive, calculation of relevance from texts of hypertext links, PDF indexing and other formats, *Google File System*, *Google Link*…
- The system anatomy. *see* [MAR]

Google: Relevance

The crucial thing is documents' relevance (credit) computation.

- ☞ Usage of tags of text and web typography for the relevance calculation of document terms.
- ☞ Usage of text of hyperlink is referring to the document.

Google: PageRank

- ***PageRank***: objective measure of page importance based on citation analysis (suitable for ordering of answers for queries, namely page relevance computation).
- Let pages $T_1, \ldots, T_n$ (citations) point to a page A, total sum of pages is m. PageRank

$$PR(A) = \frac{(1-d)}{m} + d\left(\frac{PR(T_1)}{C(T_1)} + \ldots \frac{PR(T_n)}{C(T_n)}\right)$$

- PageRank can be calculated by a simple iterative algorithm (for tens of millions of pages in hours on a normal PC).
- PageRank is a probability distribution over web pages.
- PageRank is not the only applied factor, but coefficient of more factors. A motivation with a random surfer, dumping factor d, usually around 0.85.

Data structures of Google

- Storing of file signatures
- Storing of lexicon
- Storing of hit list.
- Google File System

Index system implementation

- Inverted file – indexing file with a bit vector.
- Usage of document list to every key word.
- Coordinate system with pointers [MEL, fig. 4.18, page 46].
- Indexing of corpus texts: Finlib
  http://www.fi.muni.cz/~pary/dis.pdf *see* [MAR].
- Use of Elias coding for a compression of hit list.

Index system implementation (cont.)

- ☞ Efficient storing of index/dictionary [lemmas]: ***packed trie***, Patricia tree, and other tree structures.
- ☞ Syntactic neural network (S. M. Lucas: Rapid best-first retrieval from massive dictionaries, Pattern Recognition Letters 17, p. 1507–1512, 1996).
- ☞ Commercial implementations: *Verity engine*, most of web search engines – with few *exceptions* – hide their key to success.

Dictionary representation by FA I

Article M. Mohri: On Some Applications of Finite-State Automata Theory to Natural Language Processing see [MAR]

- Dictionary representation by finite automaton.
- Ambiguities, unification of minimized deterministic automata.
- Example: *done,do.V3:PP*
  done,done.A0
- Morphological dictionary as a list of pairs [word form, lemma].
- Compaction of storing of data structure of automata (Liang, 1983).
- Compression ratio up to 1:20 in the linear approach (given the length of word).

Dictionary representation by FA II

- ☞ Transducer for dictionary representation.
- ☞ Deterministic transducer with 1 output (subsequential transducer) for dictionary representation including one string on output (information about morphology, hyphenation,...).
- ☞ Deterministic transducer with p outputs (p–subsequential transducer) for dictionary representation including more strings on output (ambiguities).
- ☞ Determinization of the transducer generally unrealizable (the class of deterministic transducers with an output is a proper subclass of nondeterministic transducers); for purposes of natural language processing, though, usually doesn't occur (there aren't cycles).

Dictionary representation by FA III

- An addition of a state to a transducer corresponding (w_1,w_2) without breaking the deterministic property: first a state for (w_1,ε), then with resulting state final state with output w_2.
- Efficient method, quick, however not minimal; there are minimizing algorithms, that lead to spatially economical solutions.
- Procedure: splitting of dictionary, creation of det. transducers with p outputs, their minimization, then a deterministic unification of transducers and minimizing the resulting.
- Another use also for the efficient indexing, speech recognition, etc.