CE306/CE706
Information Retrieval

Ad-hoc Search & Processing Pipeline

Spring 2018
Brief Module Outline (Reminder)

- Motivation + introduction
- Processing pipeline / indexing + query processing
- Large-scale open-source search tools
- Information Retrieval models
- Evaluation
- Log analysis
- User profiles / personalisation / contextualisation
- IR applications, e.g. enterprise search
Main Information Retrieval Tasks

- Ad-hoc retrieval:
  - relatively static document collection
  - different user queries
  - examples: library catalogues, Web search engines

- Filtering/Routing:
  - relatively static user requests
  - constantly updated document collection
  - examples: news filtering, classification, spam filter …
Ad-hoc Retrieval

- Text-based retrieval
- Given a query and a corpus, find the relevant items
  - query: textual description of information need
  - corpus: a collection of textual documents
  - relevance: satisfaction of the user’s information need
- “Ad-hoc” because the number of possible queries is (in theory) infinite.
Reminder …
...or ...

Google search for ce706

About 11,100 results (0.63 seconds)

Burco Manual Fill Water Boiler 30Ltr - CE706 - Buy Online at Nisbets
www.nisbets.co.uk/burco-manual-fill-water-boiler.../CE706/ProductDetail.action

Rating: 4.8 - 8 votes
Buy Burco Manual Fill Water Boiler 30Ltr (CE706) & more from our Manual Fill Boilers range at Nisbets.co.uk. Next day delivery on thousands of catering ...

Images for ce706

ESER CE706 OR SANEMAX CE706 How to Factory Reset ... - YouTube
https://www.youtube.com/watch?v=5yt0Fw4oL5E
9 May 2014 - Uploaded by Da Irsbhoi
ESER CE706 7" HD TFT Dual Core Android 4.2 Smart Game Console w/ Wi-Fi / HDMI / Online chat - Black ...

Information Retrieval - University of Essex :: Module Directory
https://www.essex.ac.uk / modules
Ad-hoc Retrieval

• Start with ‘traditional’ document collection, i.e. treat document as plain text and not HTML document

• Look at how we match a query against a collection of documents

• To do so we need to process the document collection into an index structure that allows fast access and search
So it’s actually more like this:
Why start with plain Documents?

- Main processing/indexing pipeline is the same
- Transparent process: not constantly tweaked (e.g., links) or tuned using user-interaction data (e.g., clicks) as in Web search
- Very common: government and corporate intranets (enterprise search), social media, your own personal computers (e.g., Word and PDF documents) …
- Emergence of the Web added link structure (and more) that offers more cues to identifying relevant documents, will get back to that …
Basic Information Retrieval Process

1. Information need
2. Representation
3. Query
4. Comparison
5. Retrieved objects
6. Evaluation
7. Indexed objects
8. Representation
9. Document
Basic Information Retrieval Process

1. User's information need
2. Representation of the information need
3. Query generation
4. Comparison between the query and indexed objects
5. Retrieval of relevant objects
6. Evaluation of the retrieved objects
7. Feedback to the user
Document Representation

1. Information need
2. Representation
3. Query
4. Comparison
5. Retrieved objects
6. Evaluation
7. Indexed objects
8. Document
Most Basic View of a Search Engine

- A search engine does not scan each document to see if it satisfies the query.
- It uses an index to quickly locate the relevant documents.
- **Index**: a list of concepts with pointers to documents that discuss them.

---

*Index from Manning et al., 2008*
Most Basic View of a Search Engine

input query: A/B test

output document: docid: 170

- So, what goes in the index is important!
- How might we combine concepts, e.g.:
  ```patent search + A/B test'' ?
  ``A/B test or user test'' ? …
Most Basic View of a Search Engine

input query: A/B tests
Most Basic View of a Search Engine

input query: A/B testing
Most Basic View of a Search Engine

input query: test
Document Representation

- **Document representation**: deciding what concepts should go in the index

- **Option 1 (controlled vocabulary)**: a set of manually constructed concepts that describe the major topics covered in the collection

- **Option 2 (free-text indexing)**: the set of individual terms that occur in the collection
If we view option 1 and option 2 as two extremes, where does this particular index fit in?
Document Representation
(option 1: controlled vocabulary)

• Think of examples in which controlled vocabularies are being used
• Why would this be done?
• What are the pros and cons compared to indexing?
Document = Bag of Words
(i.e., option 2: free-text indexing)

- Need to locate important information in a document
- Deep natural language understanding too expensive
- Common assumption: document is a list of words
- Documents represented by a set of index terms
- The search engine will determine which terms are important (depends on the "retrieval model" applied)
Free-text Indexing:
Retrieval Process

1. Define text database, i.e. data sources, operations on text and text structure
2. Process document collection into index database for fast access
3. Parse user query using same text processing operations
4. Apply query operations to processed query
5. Generate system query to retrieve matching documents
6. Rank and display results
Gerard Salton

From Wikipedia, the free encyclopedia

**Gerard Salton** (6 March 1927 in Nuremberg - 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard.

Salton was born Gerhard Anton Sahlmann on March 8, 1927 in Nuremberg, Germany. He received a Bachelor's (1950) and Master's (1952) degree in mathematics from Brooklyn College, and a Ph.D. from Harvard in Applied Mathematics in 1958, the last of Howard Aiken's doctoral students, and taught there until 1965, when he joined Cornell University and co-founded its department of Computer Science.

Salton was perhaps most well known for developing the now widely used Vector Space Model for Information Retrieval. In this model, both documents and queries are represented as vectors of term counts, and the similarity between a document and a query is given by the cosine between the term vector and the document vector. In this paper, he also introduced TF-IDF, or term-frequency-inverse-document frequency, a model in which the score of a term in the a document is the ratio of the number of terms in that document divided by the frequency of the number of documents in which that term occurs. (The concept of inverse document frequency, a measure of specificity, had been introduced in 1972 by Karen Sparck-Jones.) Later in life, he became interested in automatic text summarization and analysis, as well as automatic hypertext generation. He published over 150 research articles and 5 books during his life.

Salton was editor-in-chief of the Communications of the ACM and the Journal of the ACM, and chaired SIGIR. He was an associate editor of the ACM Transactions on Information Systems. He was an ACM Fellow (elected 1995), received an Award of Merit from the American Society for Information Science (1989), and was the first recipient of the SIGIR Award for outstanding contributions to study of information retrieval (1983) – now called the Gerard Salton Award.
Gerard Salton (6 March 1927 in Nuremberg - 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard.

Salton was born Gerhard Anton Sahlimann on March 8, 1927 in Nuremberg, Germany. He received a Bachelor's (1950) and Master's (1952) degree in mathematics from Brooklyn College, and a Ph.D. from Harvard in Applied Mathematics in 1958, the last of Howard Aiken's doctoral students, and taught there until 1965, when he joined Cornell University and co-founded its department of Computer Science.

Salton was perhaps most well known for developing the now widely used Vector Space Model for information Retrieval\[^1\]. In this model, both documents and queries are represented as vectors of term counts, and the similarity between a document and a query is given by the cosine between the term vector and the document vector. In this paper, he also introduced TF-IDF, or term-frequency-inverse-document frequency, a model in which the score of a term in the a document is the ratio of the number of terms in that document divided by the frequency of the number of documents in which that term occurs. (The concept of inverse document frequency, a measure of specificity, had been introduced in 1972 by Karen Sparck-Jones\[^2\].) Later in life, he became interested in automatic text summarization and analysis\[^3\], as well as automatic hypertext generation\[^4\]. He published over 150 research articles and 5 books during his life.

Salton was editor-in-chief of the Communications of the ACM and the Journal of the ACM, and chaired SIGIR. He was an associate editor of the ACM Transactions on Information Systems. He was an ACM Fellow (elected 1995), received an Award of Merit from the American Society for Information Science (1989), and was the first recipient of the SIGIR Award for outstanding contributions to study of information retrieval (1983) – now called the Gerard Salton Award.
Gerard Salton (8 March 1927 in Nuremberg - 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard. He was editor-in-chief of the Journal of the American Society for Information Science (1989), and was the first recipient of the SIGIR Award for outstanding contributions to study of information retrieval (1983) – now called the Gerard Salton Award.
Gerard Salton (8 March 1927 in Nuremberg - 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard.
Free-text Indexing

mark-up

• Describes how the content should be presented
  ‣ e.g., your browser interprets html mark-up and presents the page as intended by the author
• Can also define relationships with other documents (e.g., hyperlinks)
• Can provide evidence of what text is important for search
• It may also provide useful, “unseen” information!
• Remember: we ignore markup for now
Gerard Salton

Gerard Salton (6 March 1927 in Nuremberg - 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard.

Salton was born Gerhard Anton Sahlimann on March 8, 1927 in Nuremberg, Germany. He received a Bachelor's (1950) and Master's (1952) degree in mathematics from Brooklyn College, and a Ph.D. from Harvard in Applied Mathematics in 1958, the last of Howard Aiken's doctoral students, and taught there until 1965, when he joined Cornell University and co-founded its department of Computer Science.

Salton was perhaps most well known for developing the now widely used Vector Space Model for information Retrieval. In this model, both documents and queries are represented as vectors of term counts, and the similarity between a document and a query is given by the cosine between the term vector and the document vector. In this paper, he also introduced TF-IDF, or term-frequency-inverse-document frequency, a model in which the score of a term in the a document is the ratio of the number of terms in that document divided by the frequency of the number of documents in which the term appears, as well as automatic hypertext generation. He published over 150 research articles and 3 books during his life.

Salton was editor-in-chief of the Communications of the ACM and the Journal of the ACM, and chaired SIGIR. He was an associate editor of the ACM Transactions on Information Systems. He was an ACM Fellow (elected 1995), received an Award of Merit from the American Society for Information Science (1989), and was the first recipient of the SIGIR Award for outstanding contributions to study of information retrieval (1983) – now called the Gerard Salton Award.

<a href="/wiki/Association_for_Computing_Machinery" >ACM</a>
Free-text Indexing
Step 1: mark-up removal

Gerard Salton (8 March 1927 in Nuremberg - 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard.

- Remove HTML mark-up
Gerard Salton (8 March 1927 in Nuremberg 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard.

- Result is plain text
Free-text Indexing
Step 2: tokenization

Gerard Salton  (8 March 1927 in Nuremberg 28 August 1995), also known as Gerry Salton, was a Professor of Computer Science at Cornell University. Salton was perhaps the leading computer scientist working in the field of information retrieval during his time. His group at Cornell developed the SMART Information Retrieval System, which he initiated when he was at Harvard.

- Splitting text into white-space separated `words`
- (Possibly) remove punctuation altogether
- Not that simple (e.g., ambiguity of punctuation)!
- i++ → i? Ph.D → Ph D? TF-IDF → TF IDF?
Free-text Indexing
Step 3: case-folding

gerard salton ( 8 march 1927 in nuremberg 28 august 1995 ), also known as gerry salton, was a professor of computer science at cornell university. salton was perhaps the leading computer scientist working in the field of information retrieval during his time. his group at cornell developed the smart information retrieval system, which he initiated when he was at harvard.

• Typical representation: lower case
• London ——> london   LONDON ——> london
• But:  SMART ——> smart
Free-text Indexing
Step 4: stop word removal

• **Stopwords**: words that we choose to ignore because we expect them to **not** be useful in distinguishing between relevant/non-relevant documents for **any** query

• Very frequent words for example (see Zipf’s law later on)

• Closed class *vs.* open class words
  (e.g. determiners and prepositions *vs.* nouns and verbs)

• But: `To be or not to be’
Free-text Indexing
Final step

- Apply the pipeline of processing steps to every document in the collection and create an index using the union of all remaining terms
- We will look at the structure of the index next week
- As we will see, different structures are possible
- Different models to match a query against the index, e.g. Boolean, Vector Space, Probabilistic … (next time)
- Note: the same processing steps need to be applied to the query that gets submitted!
Free-text Indexing
Other possible steps

• Morphological analysis, e.g. initiated —> initiate
• Stemming, e.g. computer/computing —> comput
• Part-of-speech tagging, e.g. ‘cornell/NNP university/NN’
• Identification of phrases, e.g. ‘leading computer scientist’
• Named-entity recognition, e.g. ‘cornell university’
• Relation extraction, e.g.
  is(‘gerry salton’, ‘computer scientist’)
Free-text Indexing Steps Expanded

Zipf’s Law

Probability (of occurrence)

Rank

(by decreasing frequency)
Free-text Indexing Steps Expanded

Zipf’s Law

• Distribution of word frequencies very skewed

• A few words occur very often, many words hardly ever occur, e.g., two most common words (“the”, “of”) make up about 10% of all word occurrences in text documents

• Rank \((r)\) of a word times its frequency \((f)\) is approximately a constant \((k)\), i.e.

\[f \times r = k\]
Free-text Indexing Steps Expanded

Zipf’s Law

<table>
<thead>
<tr>
<th>Word</th>
<th>Freq.</th>
<th>r</th>
<th>$P_r$ (%)</th>
<th>$r.P_r$</th>
<th>Word</th>
<th>Freq.</th>
<th>r</th>
<th>$P_r$ (%)</th>
<th>$r.P_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>2,420,778</td>
<td>1</td>
<td>6.49</td>
<td>0.065</td>
<td>has</td>
<td>136,007</td>
<td>26</td>
<td>0.37</td>
<td>0.095</td>
</tr>
<tr>
<td>of</td>
<td>1,045,733</td>
<td>2</td>
<td>2.80</td>
<td>0.056</td>
<td>are</td>
<td>130,322</td>
<td>27</td>
<td>0.35</td>
<td>0.094</td>
</tr>
<tr>
<td>to</td>
<td>968,882</td>
<td>3</td>
<td>2.60</td>
<td>0.078</td>
<td>not</td>
<td>127,493</td>
<td>28</td>
<td>0.34</td>
<td>0.096</td>
</tr>
<tr>
<td>a</td>
<td>892,429</td>
<td>4</td>
<td>2.39</td>
<td>0.096</td>
<td>who</td>
<td>116,364</td>
<td>29</td>
<td>0.31</td>
<td>0.090</td>
</tr>
<tr>
<td>and</td>
<td>865,644</td>
<td>5</td>
<td>2.32</td>
<td>0.120</td>
<td>they</td>
<td>111,024</td>
<td>30</td>
<td>0.30</td>
<td>0.089</td>
</tr>
<tr>
<td>in</td>
<td>847,825</td>
<td>6</td>
<td>2.27</td>
<td>0.140</td>
<td>its</td>
<td>111,021</td>
<td>31</td>
<td>0.30</td>
<td>0.092</td>
</tr>
<tr>
<td>said</td>
<td>504,593</td>
<td>7</td>
<td>1.35</td>
<td>0.095</td>
<td>had</td>
<td>103,943</td>
<td>32</td>
<td>0.28</td>
<td>0.089</td>
</tr>
<tr>
<td>for</td>
<td>363,865</td>
<td>8</td>
<td>0.98</td>
<td>0.078</td>
<td>will</td>
<td>102,949</td>
<td>33</td>
<td>0.28</td>
<td>0.091</td>
</tr>
<tr>
<td>that</td>
<td>347,072</td>
<td>9</td>
<td>0.93</td>
<td>0.084</td>
<td>would</td>
<td>99,503</td>
<td>34</td>
<td>0.27</td>
<td>0.091</td>
</tr>
<tr>
<td>was</td>
<td>293,027</td>
<td>10</td>
<td>0.79</td>
<td>0.079</td>
<td>about</td>
<td>92,983</td>
<td>35</td>
<td>0.25</td>
<td>0.087</td>
</tr>
<tr>
<td>on</td>
<td>291,947</td>
<td>11</td>
<td>0.78</td>
<td>0.086</td>
<td>i</td>
<td>92,005</td>
<td>36</td>
<td>0.25</td>
<td>0.089</td>
</tr>
<tr>
<td>he</td>
<td>250,919</td>
<td>12</td>
<td>0.67</td>
<td>0.081</td>
<td>been</td>
<td>88,786</td>
<td>37</td>
<td>0.24</td>
<td>0.088</td>
</tr>
<tr>
<td>is</td>
<td>245,843</td>
<td>13</td>
<td>0.65</td>
<td>0.086</td>
<td>this</td>
<td>87,286</td>
<td>38</td>
<td>0.23</td>
<td>0.089</td>
</tr>
<tr>
<td>with</td>
<td>223,846</td>
<td>14</td>
<td>0.60</td>
<td>0.084</td>
<td>their</td>
<td>84,638</td>
<td>39</td>
<td>0.23</td>
<td>0.089</td>
</tr>
<tr>
<td>at</td>
<td>210,064</td>
<td>15</td>
<td>0.56</td>
<td>0.085</td>
<td>new</td>
<td>83,449</td>
<td>40</td>
<td>0.22</td>
<td>0.090</td>
</tr>
<tr>
<td>by</td>
<td>209,586</td>
<td>16</td>
<td>0.56</td>
<td>0.090</td>
<td>or</td>
<td>81,796</td>
<td>41</td>
<td>0.22</td>
<td>0.090</td>
</tr>
<tr>
<td>it</td>
<td>195,621</td>
<td>17</td>
<td>0.52</td>
<td>0.089</td>
<td>which</td>
<td>80,385</td>
<td>42</td>
<td>0.22</td>
<td>0.091</td>
</tr>
<tr>
<td>from</td>
<td>189,451</td>
<td>18</td>
<td>0.51</td>
<td>0.091</td>
<td>we</td>
<td>80,245</td>
<td>43</td>
<td>0.22</td>
<td>0.093</td>
</tr>
<tr>
<td>as</td>
<td>181,714</td>
<td>19</td>
<td>0.49</td>
<td>0.093</td>
<td>more</td>
<td>76,388</td>
<td>44</td>
<td>0.21</td>
<td>0.090</td>
</tr>
<tr>
<td>be</td>
<td>157,300</td>
<td>20</td>
<td>0.42</td>
<td>0.084</td>
<td>after</td>
<td>75,165</td>
<td>45</td>
<td>0.20</td>
<td>0.091</td>
</tr>
<tr>
<td>were</td>
<td>153,913</td>
<td>21</td>
<td>0.41</td>
<td>0.087</td>
<td>us</td>
<td>72,045</td>
<td>46</td>
<td>0.19</td>
<td>0.089</td>
</tr>
<tr>
<td>an</td>
<td>152,576</td>
<td>22</td>
<td>0.41</td>
<td>0.090</td>
<td>percent</td>
<td>71,956</td>
<td>47</td>
<td>0.19</td>
<td>0.091</td>
</tr>
<tr>
<td>have</td>
<td>149,749</td>
<td>23</td>
<td>0.40</td>
<td>0.092</td>
<td>up</td>
<td>71,082</td>
<td>48</td>
<td>0.19</td>
<td>0.092</td>
</tr>
<tr>
<td>his</td>
<td>142,285</td>
<td>24</td>
<td>0.38</td>
<td>0.092</td>
<td>one</td>
<td>70,266</td>
<td>49</td>
<td>0.19</td>
<td>0.092</td>
</tr>
<tr>
<td>but</td>
<td>140,880</td>
<td>25</td>
<td>0.38</td>
<td>0.094</td>
<td>people</td>
<td>68,988</td>
<td>50</td>
<td>0.19</td>
<td>0.093</td>
</tr>
</tbody>
</table>

Top 50 Words from a collection of news articles (AP89)
Free-text Indexing Steps Expanded

Zipf’s Law

AP89: distribution of words (rank-ordered) on a logarithmic scale
Free-text Indexing Steps Expanded

Zipf’s Law

European Parliament: German
(transcribed speech, see http://homepages.inf.ed.ac.uk/pkoehn/)
Free-text Indexing Steps Expanded

Zipf’s Law

European Parliament: Hungarian
(transcribed speech, see http://homepages.inf.ed.ac.uk/pkoehn/)
Free-text Indexing Steps Expanded

Zipf’s Law

Alice in Wonderland
(text available via Project Gutenberg)
Free-text Indexing Steps Expanded
Morphology

- Many morphological variations of words (with same or similar meaning)
  - inflectional: ‘buy’, ‘bought’, ‘buying’
  - derivational: ‘computer’, ‘compute’, ‘computable’

- But more complicated in languages other than English, e.g.
  - Helsingin sanomat —> Helsinki +GEN + sanoma +PLURAL
  - Gaststätenneueröffnungsuntergangsgewissheit
    (Restaurant+new+opening+failure+certainty)
    (Ben Schott ``Schottenfreude: German Words for the Human Condition'', John Murray (Publishers), 2013).
Free-text Indexing Steps Expanded

Stemming

• Generally a small but significant effectiveness improvement

• Can be crucial for some languages, e.g., 5-10% improvement for English, up to 50% in Arabic

<table>
<thead>
<tr>
<th>Word</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>kitab</td>
<td>a book</td>
</tr>
<tr>
<td>kitabi</td>
<td>my book</td>
</tr>
<tr>
<td>alkitab</td>
<td>the book</td>
</tr>
<tr>
<td>kitabuki</td>
<td>your book (f)</td>
</tr>
<tr>
<td>kitabuka</td>
<td>your book (m)</td>
</tr>
<tr>
<td>kitabuhu</td>
<td>his book</td>
</tr>
<tr>
<td>kataba</td>
<td>to write</td>
</tr>
<tr>
<td>maktaba</td>
<td>library, bookstore</td>
</tr>
<tr>
<td>maktab</td>
<td>office</td>
</tr>
</tbody>
</table>

Words with the Arabic root ktb
Many queries are 2-3 word phrases

Phrases are more meaningful, less ambiguous and more precise than single words, e.g. ‘exam timetable’ vs. ‘timetable’

Often via part-of-speech (POS) tagging as the initial step

Alternatively simply word n-grams
Free-text Indexing Steps Expanded
Part-of-speech Tagging

• POS taggers use statistical models of text to predict syntactic tags of words

• Phrases can then be defined as simple noun groups, for example

Original text:
Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulate demand, price cut, volume of sales.

Brill tagger:
Free-text Indexing Steps Expanded
Part-of-speech Tagging

<table>
<thead>
<tr>
<th>TREC data</th>
<th>Patent data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>Phrase</td>
</tr>
<tr>
<td>65824</td>
<td>united states</td>
</tr>
<tr>
<td>61327</td>
<td>article type</td>
</tr>
<tr>
<td>33864</td>
<td>los angeles</td>
</tr>
<tr>
<td>18062</td>
<td>hong kong</td>
</tr>
<tr>
<td>17788</td>
<td>north korea</td>
</tr>
<tr>
<td>17308</td>
<td>new york</td>
</tr>
<tr>
<td>15513</td>
<td>san diego</td>
</tr>
<tr>
<td>15009</td>
<td>orange county</td>
</tr>
<tr>
<td>12869</td>
<td>prime minister</td>
</tr>
<tr>
<td>12799</td>
<td>first time</td>
</tr>
<tr>
<td>12067</td>
<td>soviet union</td>
</tr>
<tr>
<td>10811</td>
<td>russian federation</td>
</tr>
<tr>
<td>9912</td>
<td>united nations</td>
</tr>
<tr>
<td>8127</td>
<td>southern california</td>
</tr>
<tr>
<td>7640</td>
<td>south korea</td>
</tr>
<tr>
<td>7620</td>
<td>end recording</td>
</tr>
<tr>
<td>7524</td>
<td>european union</td>
</tr>
<tr>
<td>7436</td>
<td>south africa</td>
</tr>
<tr>
<td>7362</td>
<td>san francisco</td>
</tr>
<tr>
<td>7086</td>
<td>news conference</td>
</tr>
<tr>
<td>6792</td>
<td>city council</td>
</tr>
<tr>
<td>6348</td>
<td>middle east</td>
</tr>
<tr>
<td>6157</td>
<td>peace process</td>
</tr>
<tr>
<td>5955</td>
<td>human rights</td>
</tr>
<tr>
<td>5837</td>
<td>white house</td>
</tr>
</tbody>
</table>

Example Noun Phrases
Free-text Indexing
Some possible implementations

- Mark-up removal: e.g., jsoup, Beautiful soup
- Tokenization: regular expression matching
- Case-folding: regular expression (matching+substitution)
- Stopword removal: stopword lists
- Stemming: e.g. Porter stemmer
- Part-of-speech tagging, NER: e.g., Stanford tools
- Phrase extraction: POS tagger + rule-based POS patterns
- Relation extraction: e.g., Stanford tools, GATE
## Document Representation
controlled vocabulary vs. free-text indexing

<table>
<thead>
<tr>
<th></th>
<th>Cost of assigning index terms</th>
<th>Ambiguity of index terms</th>
<th>Detail of representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controlled Vocabularies</td>
<td>High/Low?</td>
<td>Ambiguous/Unambiguous?</td>
<td>Can represent arbitrary level of detail?</td>
</tr>
<tr>
<td>Free-text Indexing</td>
<td>High/Low?</td>
<td>Ambiguous/Unambiguous?</td>
<td>Can represent arbitrary level of detail?</td>
</tr>
</tbody>
</table>
## Document Representation

**controlled vocabulary vs. free-text indexing**

<table>
<thead>
<tr>
<th></th>
<th>Controlled Vocabularies</th>
<th>Free-text Indexing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost of assigning index terms</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Ambiguity of index terms</td>
<td>Not ambiguous</td>
<td>Can be ambiguous</td>
</tr>
<tr>
<td>Detail of representation</td>
<td>Can’t represent arbitrary detail</td>
<td>Any level of detail</td>
</tr>
</tbody>
</table>

- Both are effective and used often (think of examples!)
- We will focus on free-text indexing
  - cheap and easy
  - most search engines use it (even those that adopt a controlled vocabulary)
Document Representation

- Information need
- Query
- Representation
- Comparison
- Retrieved objects
- Evaluation
- Indexed objects
- Document representation

1
Information Need Representation

1. information need
2. representation
   - query
   - comparison
     - retrieved objects
       - evaluation

3. indexed objects
4. document
   - representation
Evaluation

information need → representation → query → comparison → retrieved objects → evaluation

document → representation → indexed objects

3
Performing Retrieval

- information need
- representation
- query
- comparison
- retrieved objects
- evaluation

- indexed objects
- document
- representation
Summary

- We focus on ad hoc retrieval
- Documents need to be indexed before they can be searched
- Two approaches to indexing: controlled vocabulary vs. free-text indexing
  - Free-text indexing is composed of sequence of typical processing steps (same steps applied to query)
  - Then compare (processed) query with index database
- Next: how do we match query against index? Ranking?
Summary

• Here is a recent example that combines controlled vocabulary and free-text indexing (read up about it and find out exactly how the two approaches are combined ... and why):

GOV.UK

Blog
Data at GDS

Organisations:  Government Digital Service

Using data science to build a taxonomy for GOV.UK

Finding Things team, 12 January 2017 — Data science

We are creating a single taxonomy for GOV.UK, to make it easier for visitors to find what they are looking for quickly. This blog is about how we have experimented with data science to save time when building taxonomies.

The Finding Things team started the process of creating a single taxonomy with pages about education. First of all we identified the topics education...
Reading

- Chapter 2
- Chapter 4.1 and 4.3
- CE706:

Acknowledgements

- Thanks again to Jaime Arguello
- Additional material as provided by Croft et al. (2015)