Text retrieval

Kristian Wichmann

February 8, 2017

1 Push and pull modes

Text retrieval happens in two major categories: push and pull.

2 Terminology

Here we describe the various ingredients needed to make a *retrieval model*:

- We work with a *vocabulary* $V = \{w_1, \cdots, w_N\}$ of words.
- A *query* q may be written $q = (q_1, \cdots, q_m)$, where $q_i \in V$.
- A *document* can be written $d_i = (d_{i1}, \cdots, d_{im_i})$.
- A *collection* of documents is $C = \{d_1, \cdots, d_M\}$.

2.1 The text retrieval problem

Given a query q, we wish to extract the set of *relevant documents* $R(q) \subseteq C$ for the query.

In practice, all we can hope for is an approximation of the relevant documents: $R'(q)$.

3 Strategies

3.1 Document selection strategies

One way to solve the text retrieval system is to build a binary classifier f, which given a document d and a query q returns either 0 or 1, depending on whether or not $d \in R'(q)$:

$$R'(q) = \{d \in C | f(d, q) = 1\} \tag{3.1}$$

Of course, any way to choose $R'(q)$ is technically a binary classifier, but the idea is that f decides the *absolute relevance* of the document - there is no further nuance beyond "yes" or "no".

3.2 Document ranking

Instead, the function f might have a continuum of real values instead of just $\{0, 1\}$. Then we might choose $R'(q)$ based on a *cutoff* θ:

$$R'(q) = \{d \in C | f(d, q) > \theta\} \tag{3.2}$$

Here f is more nuanced, and decides what is called the *relative relevance* of the document. A list of documents sorted by decreasing relevance could be constructed, and θ decides where to stop the list. Or rather, if the user browses such a list, θ is decided by the user.

Such a list is (under certain conditions) guaranteed by the *probability ranking principle* (PRP) to be of optimal utility to the user.

4 Building a selection function f

4.1 Similarity-based models

4.1.1 Vector space model

4.2 Probabilistic models

4.3 Probabilistic inference models

4.4 Axiomatic models