Indexing

March 20, 2015

1 Documents and query representation

* term incidence matrix
* About retrieval Models

1.1 bag of words representation

* TF, DF, DLength, AVG(DLength), V, N, IDF
* subsection what and how to get from index

2 Preprocessing

In Information Retrieval, it is often necessary to interpret natural text where a a large amount of text has to be interpreted, so that it is available as a full text search and is represented efficiently in terms of both space (document storing) and time (retrieval processes) requirements.

It can also be regarded as : process of incorporating a new document into an information retrieval system.

2.1 Tokenization

Tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. Input: ”John_DavenPort #person 52 years_old #age”

John | DavenPort | person | 52 | years | old | age

2.2 Stopwords

Stopwords refer to the words that have no meaning for ”Retrieval Purposes”. E.g.

- **Articles** : a, an, the, etc.
- **Prepositions** : in, on, of, etc.
- **Conjunctions** : and, or, but, if, etc
- **Pronouns** : I, you, them, it, etc
- **Others** : some verbs, nouns, adverbs, adjectives (make, thing, similar, etc.).

Stopwords can be up to 50% of the page content and not contribute to any relevant information w.r.t. retrieval process. Removal of these can improve the size of the index considerably. Sometimes we need to be careful in terms of words in phrases! e.g.: Library of Congress, Smoky the Bear!

Word	Occurrences	Percentage
the	8,543,794	6.8
of	3,893,790	3.1

Q: There's more to these word/occurrences, if the format looks OK, I'll add?

2.3 Stemming

Stemming is commonly used in Information Retrieval to conflate morphological variants. Typical stemmer consists of collection of rules! and/or dictionaries! Similar approach may be used in other languages too!

e.g.: The following stem to the word as shown below:

servomanipulator	← servomanipulators servomanipulator
logic	← logical logic logically logics logicals logicial logicially
login	← login logins
microwire	← microwires microwire
knead	← kneaded kneads knead kneader kneading kneaders

2.3.1 Porter Stemmer

Q: should I add Porter stemmer desc?

2.3.2 Stemming Example

Original text	**Porter Stemmer**
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	market strateg carr compan agricultur chemic report predict market share chemic report market statist agrochem pesticid herbicid fungicid insecticid fertil predict sale stimul demand price cut volum sale

2.4 Term Positions

3 Index Construction

A reasonably-sized index of the web contains many billions of documents and has a massive vocabulary. Search engines run roughly 105 queries per second over that collection. We need fine-tuned data structures and algorithms to provide search results in much less than a second per query. O(n) and even O(log n) algorithms are often not nearly fast enough. The solution to this challenge is to run an inverted index on a massive distributed system.

Text search has unique needs compared to, e.g., database queries, and needs its own data structures – primarily, the inverted index.

- **Forward Index** : A forward index is a map from documents to terms (and positions). These are used when you search within a document.
- **Inverted Index** : An inverted index is a map from terms to documents (and positions). These are used when you want to find a term in any document.

3.1 Inverted lists and catalog/offset files

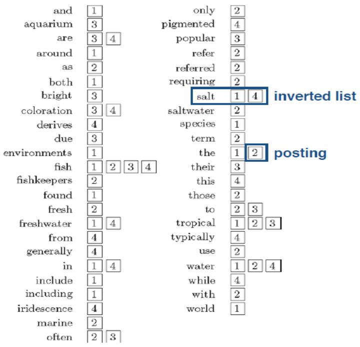
In an inverted index, each term has an associated inverted list.

At minimum, this list contains a list of identifiers for documents which contain that term.

Usually we have more detailed information for each document as it relates to that term. Each entry in an inverted list is called a posting.

Document postings can store any information needed for efficient ranking.

For instance, they typically store term counts for each document – tfw,d. Depending on the underlying storage system, it can be expensive to increase the size of a posting. It's important to be able to efficiently scan through an inverted list, and it helps if they're small.



and	1			
aquarium	3			
are	3	4		
around	1			
as	2			
both	1			
bright	3			
coloration	3	4		
derives	4			
due	3			
environments	1			
fish	1	2	3	4
fishkeepers	2			
found	1			
fresh	2			
freshwater	1	4		
from	4			
generally	4			
in	1	4		
include	1			
including	1			
iridescence	4			
marine	2			
often	2	3		
only	2			
pigmented	4			
popular	3			
refer	2			
referred	2			
requiring	2			
salt	1	4		
saltwater	2			
species	1			
term	2			
the	1	2		
their	3			
this	4			
those	2			
to	2	3		
tropical	1	2	3	
typically	4			
use	2			
water	1	2	4	
while	4			
with	2			
world	1			

Simple Inverted Index

3.2 Memory Structure, and limitations

3.3 option1: Multiple Passes

3.4 option2: Partial inverted lists

3.5 option3: preallocate the right amount of space

3.6 Updating an inverted index

4 Other things to store in the index

5 Proximity Search

virgil

$\sum_{i=1}^{5} a_i$

6 Compression

*probabilities as matching evidence

6.1 Basics of Compression, Entropy

6.2 Restricted Variable-Length Codes

6.3 Huffman codes

6.4 Lempel Ziv

7 Encoding integers

8 Distributed Indexing

9 Map-Reduce

10 Big Table

11 Query Processing