Lesson 6

Inverted Indexing for Text Retrieval

The three in one structure of the Unified Field

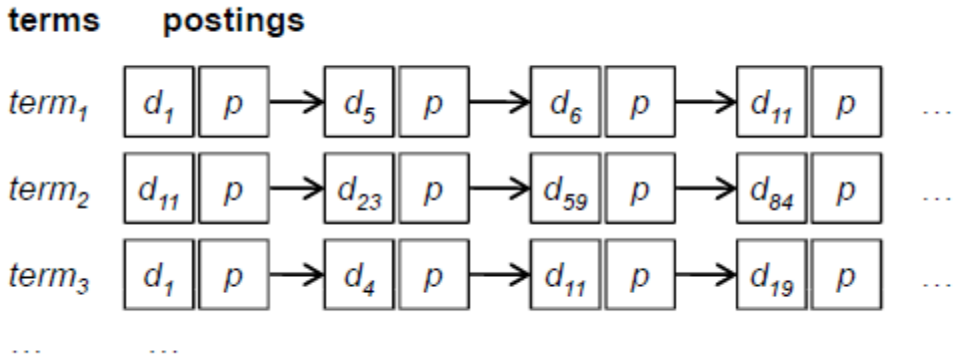
WHOLENESS OF THE LESSON

Nearly all retrieval engines for full-text search today rely on a data structure called an inverted index, which given a term provides access to the list of documents that contain the term.

Rishi (knower), Devata (process of knowing) and Chhandas (known) are the three basic qualities that structure natural law.

INVERTED INDEXES

- An inverted index is a collection of **postings lists**.
- One posting list for each “term”.



- A **posting** consists of docId and **payload**.
- The simplest payload is “nothing”.
- In our example, payload is **term frequency**.
- Term frequency is the number of times the term appear in the doc.

INVERTED INDEXES

- Postings are kept in sorted order. In our example, they are kept in sorted by docId.
- The document ids have no inherent semantic meaning, although assignment of numeric ids to documents need not be arbitrary.
- Pages from the same domain may be consecutively numbered.
- Alternatively, pages that are higher in quality (based on PageRank values) might be assigned smaller numeric values.
- An auxiliary data structure is necessary to maintain the mapping from integer document ids to some other more meaningful handle, such as a URL.

INVERTED INDEXES

- Given a query, retrieval involves:
  - fetching postings lists associated with query terms
  - In the simplest case, boolean retrieval involves set operations (union for boolean OR and intersection for boolean AND) on postings lists.

INVERTED INDEXING

```
class Mapper
    method Initialize
        H ← new AssociativeArray
        id = 0
    method Map(docid n, doc d)
        id = n
        for all term t in record r do
            H{t} ← H{t} + 1
    method close
        for all term t in H do
            Emit((t, id), H{t})
```

INVERTED INDEXING

```
class Reducer
    method Initialize
        tprev ← Ø, P ← new PostingsList
    method Reduce((t, n), [f])
        if (t ≠ tprev && tprev ≠ Ø) then
            Emit(term tprev, postings P),    P.Reset()
        P.Add(new Pair(n, f)), tprev ←  t
    method Close
        Emit(term tprev; postings P)
```

Main Point 1

An inverted index consists of postings lists, one associated with each term that appears in the collection. Thus an inverted index is a linked list of postings. *A graphical technique employed by Vedic science is the unified field chart which gives a holistic overview of a discipline and links all knowledge with the Self.*

INDEX COMPRESSION

- Each posting consists of a document id and the term frequency.
- document id can be represented as a 32-bit integer and the second as a 16-bit integer.

  [(3, 2), (7, 3), (12, 1), (49, 1), (55, 2), …],

- All brackets, parentheses, and commas are only included to enhance readability; in reality the postings would be represented as a long stream of integers.

Step 0: Common to all compression schemes

[(3, 2), (7, 3), (12, 1), (49, 1), (55, 2), …]

is represented with d-gaps as follows:

[(3, 2), (4, 3), (5, 1), (37, 1), (6, 2), …]

4 = 7 – 3

5 = 12 – 7

37 = 49 – 12

6 = 55 – 49

Stores the following sequence of non-zero integers:

3 2 4 3 5 1 37 1 6 2

INDEX COMPRESSION

We will study three different data compression techniques:

- Byte-Aligned
- Word-Aligned
- Bit-Aligned

BYTE-ALIGNED CODES

- use as many bytes as is necessary to represent the integer.
- This is known as **variable-length integer coding** (**varInt** for short) and accomplished by using the high order bit of every byte as the continuation bit, which is set to one in the last (lowest) byte and zero elsewhere.

BYTE-ALIGNED CODES

Example

127

11111111

128

00000001 10000000

WORD-ALIGNED CODES

- Simple-9, based on 32-bit words.
- Four bits in each 32-bit word are reserved as a selector.
- The remaining 28 bits are used to code actual integer values.
- There are a 9 of ways these 28 bits can be divided to code one or more integers and hence the name: Simple -9

WORD-ALIGNED CODES

Simple-9

Selector	Code length	Number of codes	Bits unused
0000	1	28	0
0001	2	14	0
0010	3	9	1
0011	4	7	0
0100	5	5	3
0101	7	4	0
0110	9	3	1
0111	14	2	0
1000	28	1	0

BIT-ALIGNED CODES

Elias-γ

An integer $x > 0$ is broken into two components, $1 + \lfloor \log_2 x \rfloor$ (= n, the length), which is coded in unary code, and $x - 2^{\lfloor \log_2 x \rfloor}$ (= r, the remainder), which is in binary. The unary component n specifies the number of bits required to code x, and the binary component codes the remainder r in n - 1 bits.

BIT-ALIGNED CODES

Example

x = 10.

$1 + \lfloor \log_2 10 \rfloor = 4$, which is 1110 in unary code.

The binary component codes $x - 2^{\lfloor \log_2 10 \rfloor} = 10 - 8 = 2$ is coded in 4 - 1 = 3 bits, which is 010.

Putting both together, we arrive at 1110010.

BIT-ALIGNED CODES

Easy Trick by Dr. Nair:

10 = 1010 in binary. Thus there are FOUR bits in the binary representation.

So start with FOUR – 1 = 3 one's.

111

Then a zero

1110

Then last FOUR – 1 binary digits

1110010

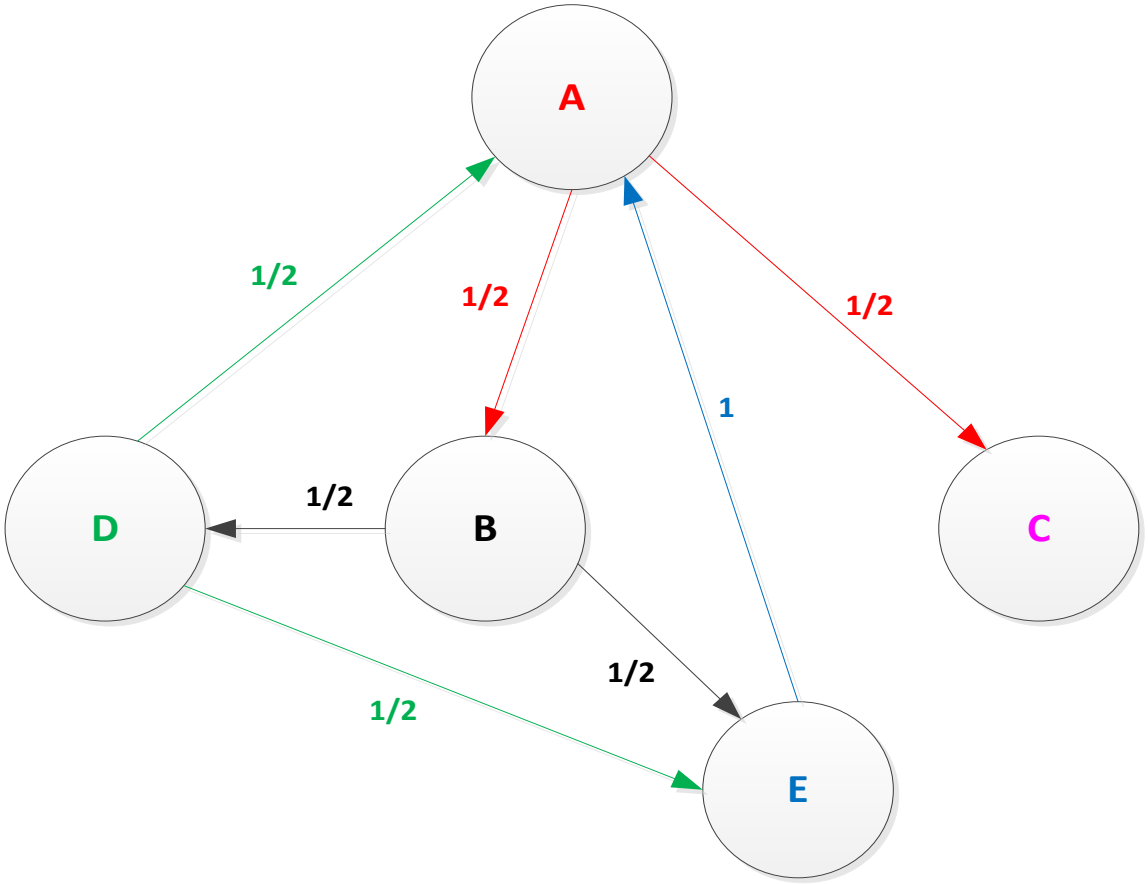
Page Rank Algorithm

Suppose the number of webpages is n.

First, we initialize the page rank of all webpages with 1/n.

Then step by step we calculate Page Rank for each Webpage one after the other.

Example



There are 5 Web pages represented by Nodes A, B, C, D and E. The hyperlink from each webpage to the other is represented by the arrow head.

Initialization

At 0th Step we have all Webpages PageRank values 0.2 that is 1/5 (1/n).

Steps	A	B	C	D	E
0	0.2	0.2	0.2	0.2	0.2

Iteration step 1.

To get PageRank of Webpage A, consider all the incoming links to A. So we have half the Page Rank of D (since D has two out links, A get 1/2 of D's page rank) and full Page Rank of E (since E has one out link, A get 1/1 of E's page rank). So it will be (1/5)*(1/2) + (1/5)*(1/1) which is (3/10) or 0.3 the Page Rank of A.

Iteration step 1 (continued)

Similarly the Page Rank of B will be (1/5)*(1/2) which is (1/10) or 0.1 because A's PageRank value is **1/5** or **0.2** from Step 0 . Even though we got 0.3 of A's PageRank in Step 1 we will not use it in this Step 1. We will use it only in the next Step.

In a similar way we calculate all the Page Rank Values.

Steps	A	B	C	D	E
0	0.2	0.2	0.2	0.2	0.2
1	0.3	0.1	0.1	0.1	0.2

We consider (N-1)th step values when we are calculating the Page Rank values for Nth Step.

Steps	A	B	C	D	E
0	0.2	0.2	0.2	0.2	0.2
1	0.3	0.1	0.1	0.1	0.2
2	0.25	0.15	0.15	0.05	0.1

We will do it till there is no change. Then we will sort them to get the most important webpages to be displayed in the search results.

Main Point 2

A simple approach to compression is to use as many bytes as is necessary to represent the integer. This is known as variable-length integer coding (varInt for short) and accomplished by using the high order bit of every byte as the continuation bit, which is set to one in the last (lowest) byte and zero elsewhere. *Vedic Science locates the source of all diversity in the non-changing Unified Field.*

CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

1. The inverted index model employs three basic concepts: term, documents, and relationship.

2. An inverted index technique is exclusively used by all web browsers.

---

3. ***Transcendental consciousness:*** *is the experience of the simplest and most abstract state of awareness which underlies all states of greater excitation.*

4. ***Impulses within the Transcendental Field:*** *Nature accomplishes what it needs by having its impulses in the transcendental field be as efficient as possible.*

5. ***Wholeness moving within itself:*** *In unity consciousness one experiences that all layers of the universe are only different expressions of the same infinite field of pure consciousness..*