# The term vocabulary and postings lists

lightsilver

Nov 9th

# Obtain Character Sequence

- Convert byte sequence into a linear sequence of characters.
  - Complex to determine the encoding
    - Machine learning
    - Heuristic
    - User select
    - Using metadata
  - Complex according to file format
    - Markup in XML
    - Licensing software library

# Document Unit

- File interface
  - Spilt attachments from mails
  - Combine separate HTML pages into single PPT
- Granularity level
  - Large: Get spurious matches, rare relevant result
  - Small: Miss important passage between documents
  - Precision / Recall tradeoff

# Tokenization

Input: Friends, Romans, Countrymen, lend me your ears;

Output: | Friends | Romans | Countrymen | lend | me | your | ears |

- Simply spilt by white space and punctuation?
  - Larger vocabulary
  - Language
  - …

# Who is most likely correct 1.0

- O’Neil
  - neill
  - oneill
  - o’neil
  - o’ & neil
  - o & neil
- Aren’t
  - aren’t
  - arent
  - are & n’t
  - aren & t
- Type
  - class of all tokens containing the same character sequence.

# Typed issue

- What if do the exact same tokenization of document and query words, generally by processing queries with the same tokenizer.

# Typed issue

- What if do the exact same tokenization of document and query words, by processing queries with the same tokenizer.

- Always matches using postings lists?

# Who is most likely correct 2.0

- C#
- bgisgood@gmail.com
- 10.13.122.223
- 1Z9999W99845399981
- http://mail.mstczju.org

- Semantic partition
  - Expend vocabulary / Restrict query input tradeoff
  - Metadata

# Who is most likely correct 3.0

- Lebensversicherungsgesellschaftsangestellter
  - subdivided into multiple words
- 莎拉波娃现在居住在美国东南部
  - -_-
- 和尚
  - Monk or and + still?

- Language identification
  - word segmentation

# Stop words

- Of little value in helping select documents matching a user need are excluded from the vocabulary entirely.
- Determined by collection frequency.
  - > Stop list

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| a | an | and | are | as | at | be | by | for | from |
| has | he | in | is | it | its | of | on | that | the |
| to | was | were | will | with | | | | | |

# Who is most likely correct 3.0

- “President of the United States”
  - “President” AND “United States”
- “To be or not to be”
  - Empty query. -_-

- Trend
  - Smaller stop list(precise > efficiency)

# Normalization

- Canonicalize tokens despite superficial differences
- Equivalence Class
  - remove characters
    - anti-discriminatory and antidiscriminatory
  - maintain relations between unnormalized tokens
    - car and automobile
      - Index expansion(space)
      - Query disjunction(time)
  - pluskid / p1uskid

# Who is most likely correct 4.0

- | Query term | Terms in documents that should be matched |
  | --- | --- |
  | Windows | Windows |
  | windows | Windows, windows, window |
  | window | window, windows |
- USA & U.S.A
  - ok
- C.A.T & cat
  - =_=
- naive & naïve
- 3/12/91
  - US Mar. 12, 1991, whereas in Europe it is 3 Dec 1991.

# Semming&Limmatization

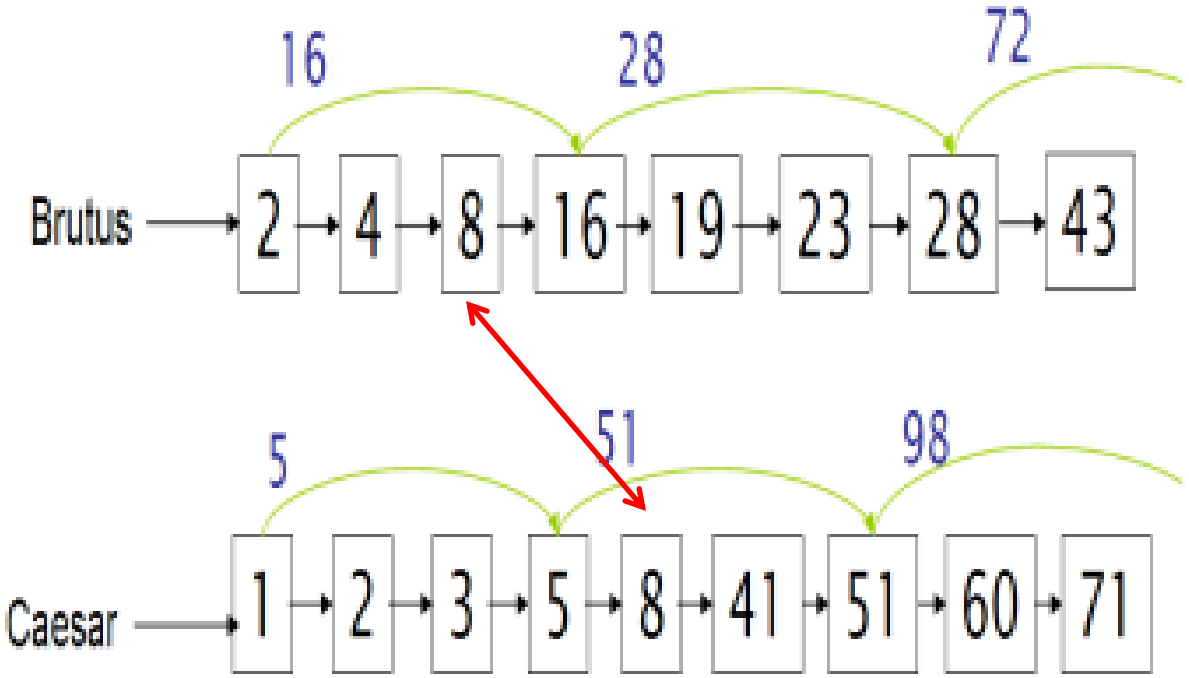
- Reduce inflectional forms and derivationally related forms of a word to a common base form.
- the boy's cars are different colors ⇒
- the boy car be differ color

- Plug-in processing index.
- Porter's algorithm→Semming

# Who is most likely correct 5.0

- Stemming increases recall while harming precision.
- Porter semmer:
  - operate operating operates operation operative operatives operational
  → oper
- Lemmatizer:
  - operating system → operate system

# Skip Pointer




- Only for original postings lists.
- Only for “and” queries.

# Skip Pointer cont.

- Where to place skip pointers?→Tradeoff
  - More skips — Fewer skips
  - Shorter spans — Fewer pointer comparisons
  - More comparisons — Longer spans
  - Space occupancy — Fewer opportunity to skip
  - P^(1/2) evenly-spaced skip pointers from practice

- What about updates?

# Phrase Query

- Double quotes syntax (“stanford university”) doesn’t match this.
  - The inventor Stanford Ovshinsky never went to university.
- Proximity Weight Ranked Retrieval ?
- Biword index
- Positional index

# Biword Index

- “stanford university palo alto” →
  - “stanford university” & “university palo” & “palo alto”

- “renegotiation of the constitution”
- →NX*N
- →NN
- →”renegotiation constitution”

# Who is most likely correct 6.0

- “cost overruns on a power plant”
  →“cost overruns”
  →“overruns power” → omit this?
  →“power plant”

→Long phrases storage size

→Less recall

# Positional Index

- Frequency for weighting

to, 993427:
⟨ 1, 6: ⟨7, 18, 33, 72, 86, 231⟩;
2, 5: ⟨1, 17, 74, 222, 255⟩;
4, 5: ⟨8, 16, 190, 429, 433⟩;
5, 2: ⟨363, 367⟩;
7, 3: ⟨13, 23, 191⟩; . . . ⟩

- Further restrict the list of possible candidates
- Check compatible positions of appearance
- “to be or not to be”

to: ⟨. . . ; 4:⟨. . . ,429,433⟩; . . . ⟩
be: ⟨. . . ; 4:⟨. . . ,430,434⟩; . . . ⟩

# Algorithm

```
POSITIONALINTERSECT(p1, p2, k)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then l ← ⟨ ⟩
 5             pp1 ← positions(p1)
 6             pp2 ← positions(p2)
 7             while pp1 ≠ NIL
 8             do while pp2 ≠ NIL
 9                do if |pos(pp1) − pos(pp2)| ≤ k
10                      then ADD(l, pos(pp2))
11                      else if pos(pp2) > pos(pp1)
12                              then break
13                   pp2 ← next(pp2)
14                while l ≠ ⟨ ⟩ and |l[0] − pos(pp1)| > k
15                do DELETE(l[0])
16                for each ps ∈ l
17                do ADD(answer, ⟨docID(p1), pos(pp1), ps⟩)
18                pp1 ← next(pp1)
19             p1 ← next(p1)
20             p2 ← next(p2)
21        else if docID(p1) < docID(p2)
22                then p1 ← next(p1)
23                else p2 ← next(p2)
24  return answer
```

- Position w/o compression
- Complexity
- BI PI combined
  - “Michael Jackson”
  - “Britney Spears”
  - “The Who”

- Improvement: Next Word Index