INFORMATION RETRIEVAL

Week 3 – Term vocabulary

Severin Mills

Who am I?

- Severin Mills
- 4th Semester BSc
- Usually in HG F 26.5
- Website: smills.ch
  - Slides, Material will be uploaded here
- E-Mail: smills@ethz.ch

Today

1

Exercise Recap

- Discussion
- Questions

2

Theory

- Tokenization
- Linguistic preprocessing
- Skip lists
- Advanced indices

3

Kahoot / Exam questions

- Exercise 2: Advanced indices
- Vote: Kahoot or Exam questions

Discussion

A conjunctive query, e.g. "**Brutus AND Caesar**" can be be evaluated in O(x+y) time, where x and y are the lengths of the posting lists for Brutus and Caesar. The same holds for disjunction e.g. "**Brutus OR Caesar**". Is this still true for "**Brutus AND NOT Caesar**"?

A conjunctive query, e.g. "**Brutus AND Caesar**" can be evaluated in O(x+y) time, where x and y are the lengths of the posting lists for Brutus and Caesar. The same holds for disjunction e.g. "**Brutus OR Caesar**". Is this still true for "**Brutus OR NOT Caesar**"?

Discussion

Question:

The expected size of the intermediate results (as indicated by the length of the postings list) is a good heuristic for deciding in which order to evaluate a query with multiple operators e.g. "**friends AND romans AND countrymen**". Suppose that one of the conjuncts is negated e.g. "**friends AND romans AND NOT countrymen**". How could we use the size of the postings list for countrymen to determine the optimal query order?

Answer:

The result of **NOT y** is the size of the domain (the number of documents in the corpus), minus the length of the postings list for y. Thus we should treat **NOT y** the opposite way to however we treat y: if the postings list is small compared with the domain, then **NOT y** will likely be large, and is thus best evaluated last, so that the result is pruned by the (hopefully) small result of the rest of the query. The size of the negated query depends on both the length of the postings list, and on the size of the corpus, while that of the non-negated only on the postings list.

Discussion

How should a (sub)query of the form "x AND NOT y" be handled in general? Fill the gaps in the following algorithm:

Algorithm:

```
intersect_not(p1, p2)
  answer <- {}
  while [          ] do
    if p2 = NIL or [          ] then
      add(answer, docID(p1))
      [          ]
    else if [          ] then
      p1 <- next(p1)
      p2 <- next(p2)
    else
      [          ]
  return answer
```

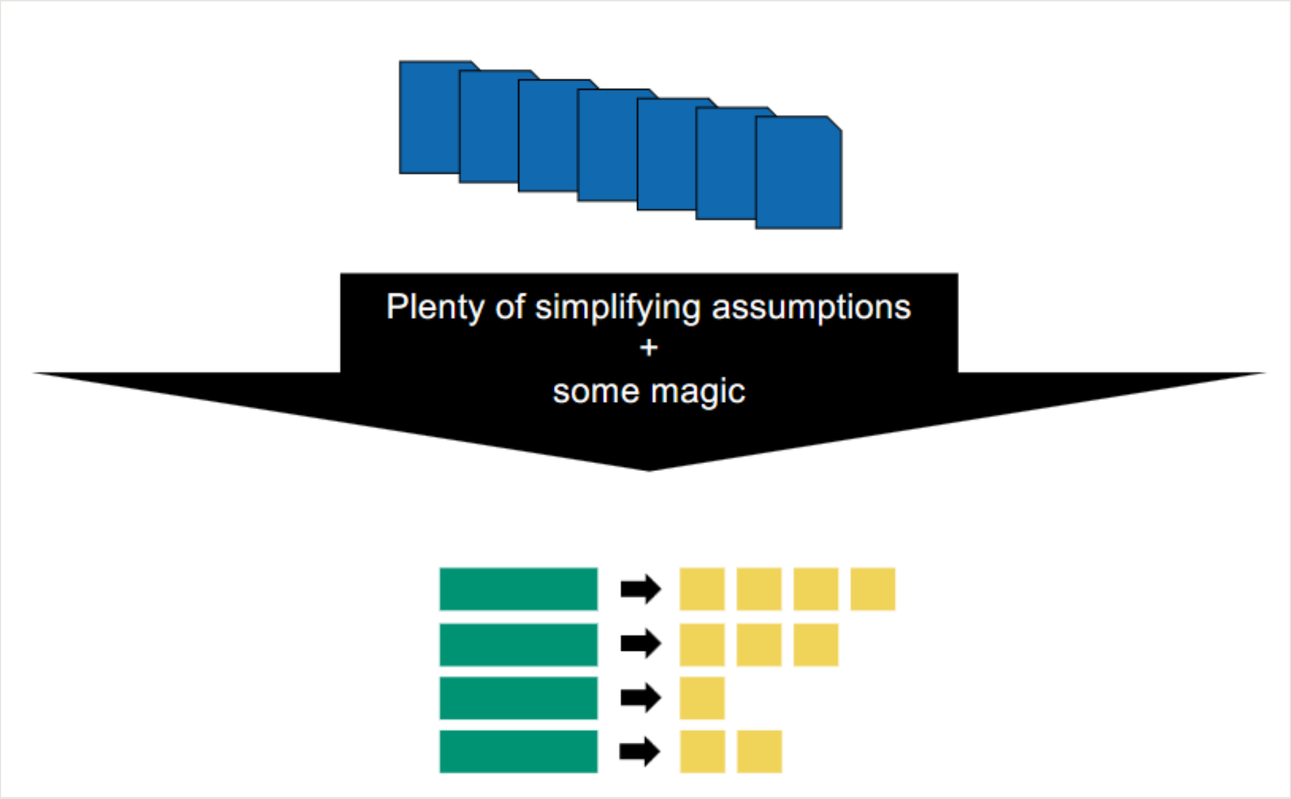
- p1 != nil
- p2 ← next(p2)
- docID(p1) = docID(p2)
- p1 ← next(p1)
- docID(p1) < docID(p2)

Discussion

- Moodle Quizzes are very helpful for exam – do them!
- Questions to Quiz or Notebooks?

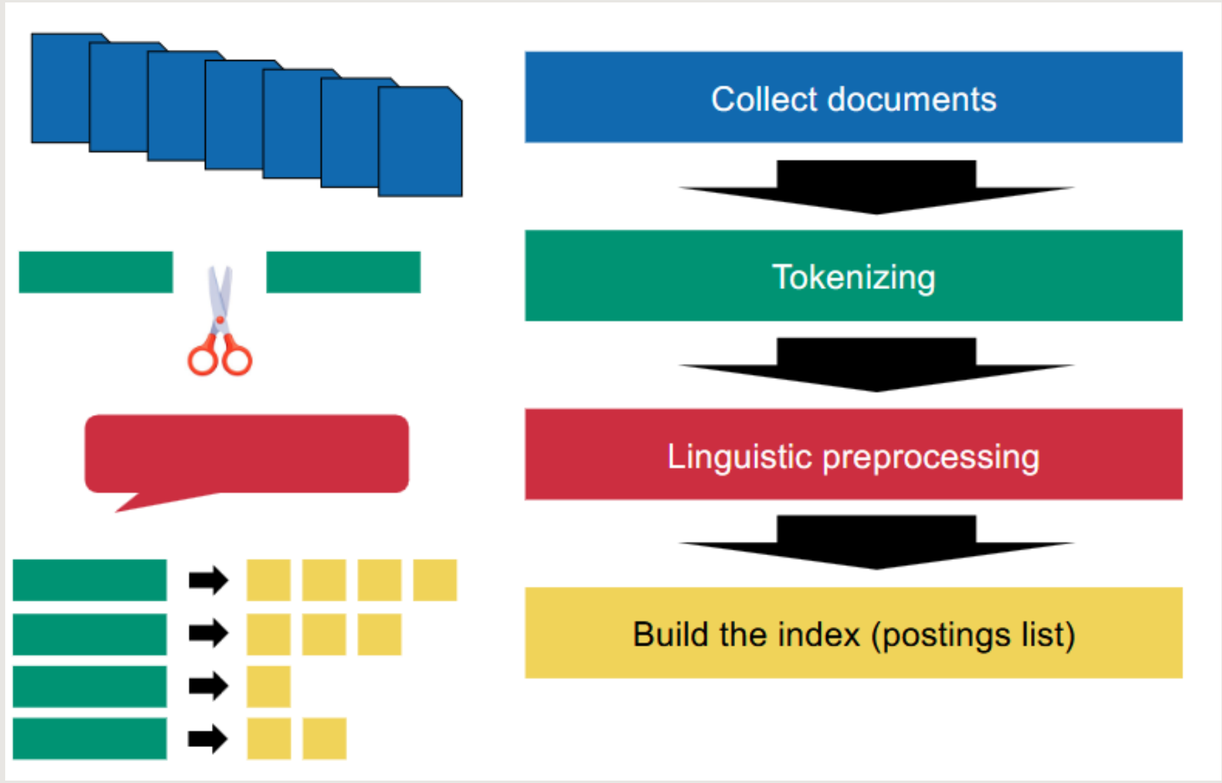
Index construction

- Previously:



Index construction

- Actually:



Collecting documents

- What is a document?
- Language?
- Encoding?
- Context?

Tokenization

- Throw away punctuation, symbols
  - Issues?
- Remove stop words: is, the, as, from (Reuters' list)
  - Why?

Corner cases!

Hewlett-Packard
State-of-the-art
co-education
the hold-him-back-and-drag-him-away maneuver
data base
San Francisco
Los Angeles-based company
cheap San Francisco-Los Angeles fares York University vs. New York University

Tokenization

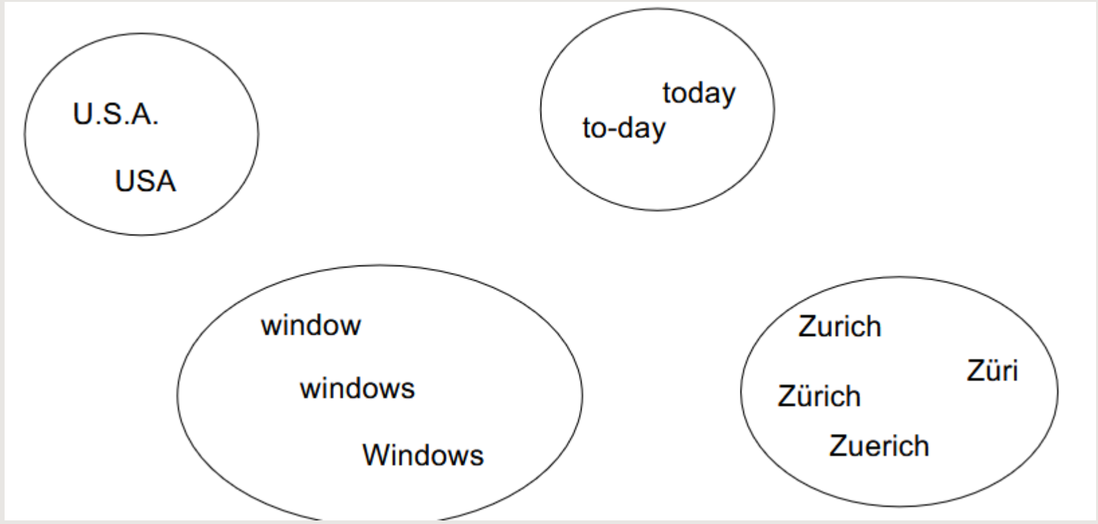
- Token vs. Type
  - e.g. “Big dogs love big dogs”
  - Tokens: “Big”, “dogs”, “love”, “big”, “dogs”
  - Types (terms): “Big”, “dogs”, “love”
- Normalized vs. Non-normalized

Raw or processed	Tied to document	Full name	Simplified/casual
raw	tied with position	positional token	token (implicitly positional)
raw	tied without position	non-positional token	
raw	not tied	word, non-normalized type	type (implicitly non-normalized in the book) token (compiler community)
processed	tied with position	positional posting	
processed	tied without position	non-positional posting	posting (implicitly non-positional)
processed	not tied	normalized type, term (if in index)	

Learn by heart for exam!

Linguistic preprocessing

- Normalization
  - Equivalence classes
  - Lowercasing (“Apple” -> “apple”)
  - Removing characters (“U.S.A.” -> “USA”)
  - Accents and diacritics
- Expansion (indexing vs. querying)
  - “lift”, “elevator”



Stemming

- Simple Rules for dropping letters
- Meaning of word not considered
- Porter Stemmer

Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

Porter
such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

Lovins
such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpres

Paice
such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

Lemmatization

- Morphological analysis of words
- Meaning of word considered
- e.g. better -> good

I **would** like a coffee
You **would** like a coffee
He **would** like a coffee
We **would** like a coffee
You **would** like a coffee
They **would** like a coffee

Je **voudrais** un café
Tu **voudrais** un café
Il/elle **voudrait** un café
Nous **voudrions** un café
Vous **voudrienz** un café
Ils/elles **voudraient** un café

Jag **skulle vilja ha** en kaffe
Du **skulle vilja ha** en kaffe
Han **skulle vilja ha** en kaffe
Vi **skulle vilja ha** en kaffe
Ni **skulle vilja ha** en kaffe
De **skulle vilja ha** en kaffe

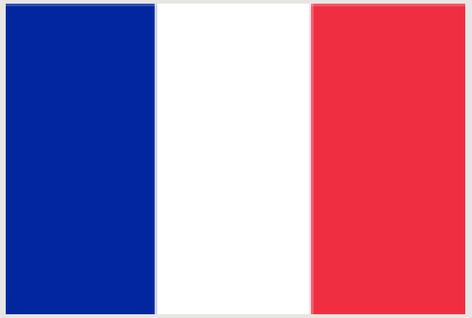
मुझे एक कॉफी चाहिए
तुमको एक कॉफी चाहिए
उसे एक कॉफी चाहिए
हमें एक कॉफी चाहिए
आपको एक कॉफी चाहिए
उन्हें एक कॉफी चाहिए

Stemming vs. Lemmatization

- What to use when?

Stemming

Lemmatization

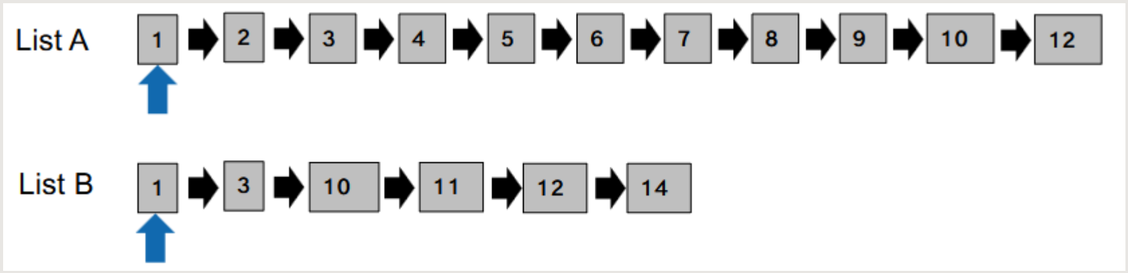


Lemmatization

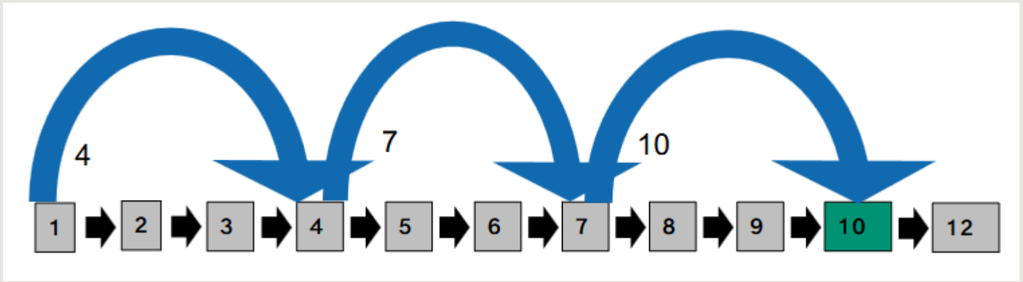
=> Depends mostly on amount of morphology

Skip Lists

- Intersection algorithm:



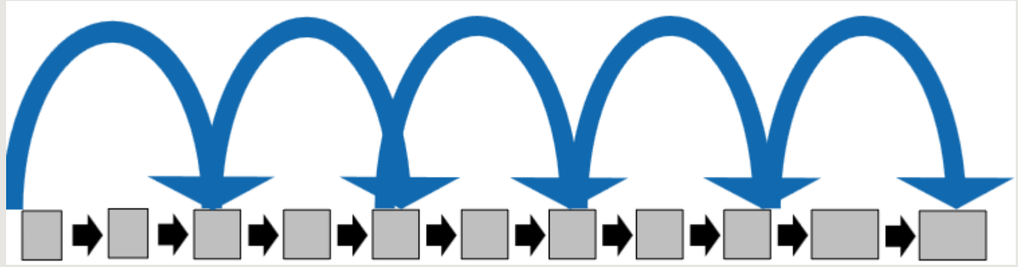
- What happens if size of postings list is huge?
- Possible Solution: Skip Lists!



Skip Lists

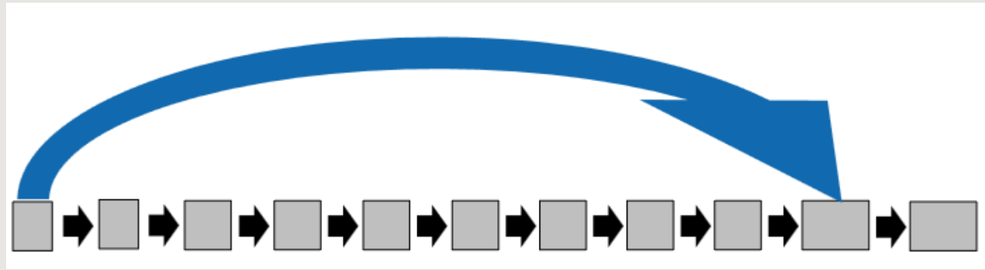
- Too short skips?

➢ ineffective
➢ many comparisons
➢ waste of space

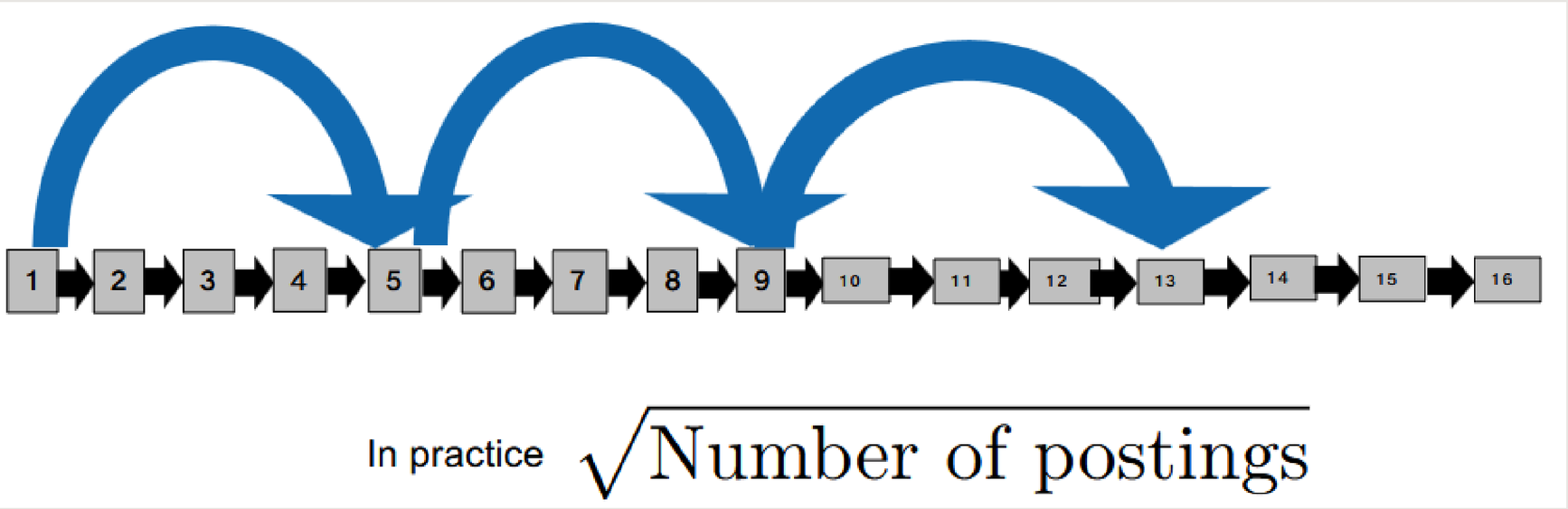


- Too long skips?

➢ too few comparisons
➢ not very usable (few opportunities)



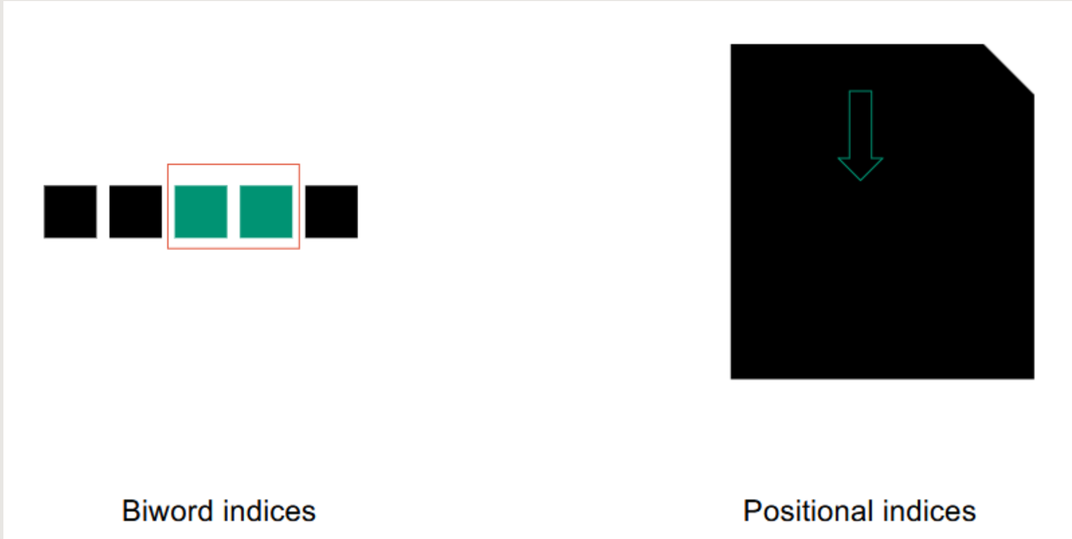
Skip Lists



Phrase search

- In the real world: one-word queries are uncommon
- “Information Retrieval” instead of “Information” AND “Retrieval”
- Inverted index cannot be used directly for phrase queries
  - Why?

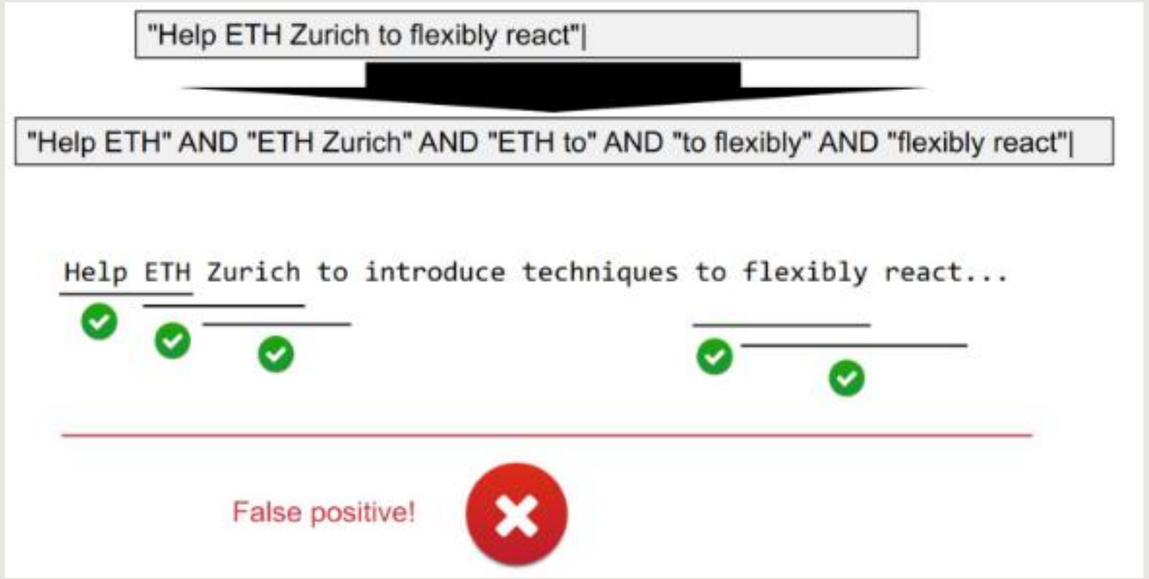
⇒ No information on proximity!
Solutions?



Bi-word Index (alt. k-word Index)

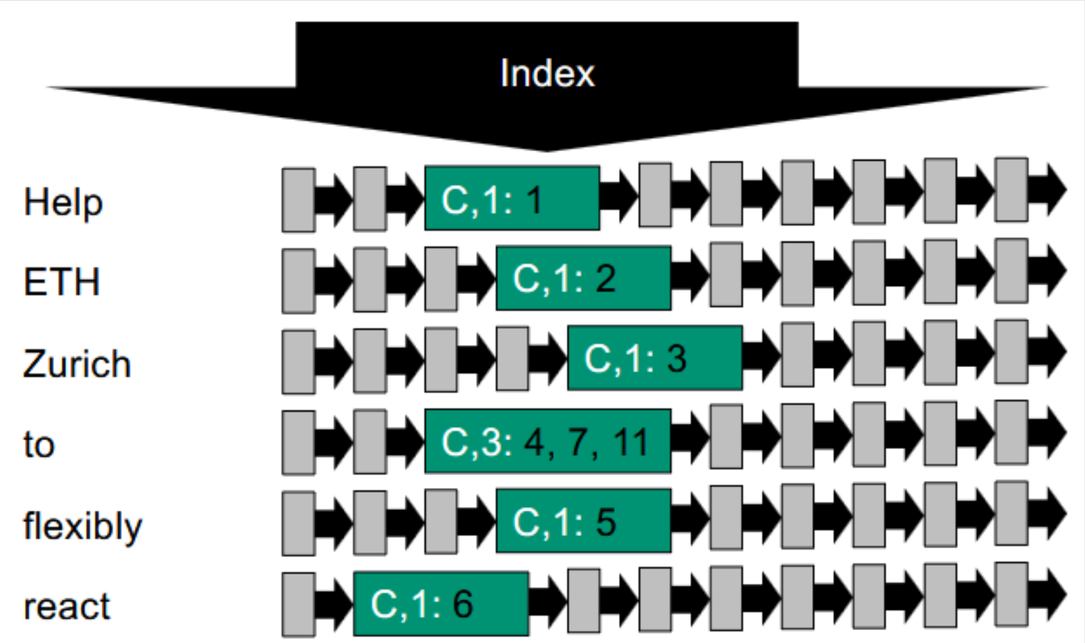
- Disadvantages?

- False positives
- Exponential vocabulary increase (#terms)^k



Positional Index

- Store docID, term frequency and position in document



Advanced indices

- Notebook
  - k-word index
  - Positional Index
  - Stemming
- Moodle quiz
  - Using the Porter Stemmer

Exam questions or Kahoot?

https://create.kahoot.it/details/cbb1ed53-a726-44d0-ab70-0a976c7b6505

FS22 – Question 12

Due to a natural catastrophe, some information was lost in the data center in which a library stored its documents and inverted indexes

In particular, they lost some parts of the document with ID 2. They now only know that it says:

... likes to ... and ... likes .....

Luckily, it was recorded in a positional index. But unfortunately, they also lost some parts of the index and only have the lists for the following terms:

she: **1**: [1, 5], **2**: [1], **3**: [3], **4**: [3], **5**: [1]

eat: **3**: [8], **2**: [4], **5**: [8]

pink: **4**: [8], **5**: [7]

recipe: **3**: [2]

cook: **1**: [4], **5**: [4]

Which document could potentially be document 2?

Note: We assume that the first word in the document is automatically capitalized and that full stop is added at the end of the document.

She likes to eat and he likes to eat.

FS22 – Question 12

Due to a natural catastrophe, some information was lost in the data center in which a library stored its documents and inverted indexes

In particular, they lost some parts of the document with ID 2. They now only know that it says:

... likes to ... and ... likes .....

Luckily, it was recorded in a positional index. But unfortunately, they also lost some parts of the index and only have the lists for the following terms:

she: **1**: [1, 5], **2**: [1], **3**: [3], **4**: [3], **5**: [1]

eat: **3**: [8], **2**: [4], **5**: [8]

pink: **4**: [8], **5**: [7]

recipe: **3**: [2]

cook: **1**: [4], **5**: [4]

Which document could potentially be document 2?

Note: We assume that the first word in the document is automatically capitalized and that full stop is added at the end of the document.

She likes to eat and he likes to eat.

The postings list for “eat” only contains index 4, but in the statement, it is also used in position 9.

FS22 – Question 12

Due to a natural catastrophe, some information was lost in the data center in which a library stored its documents and inverted indexes

In particular, they lost some parts of the document with ID 2. They now only know that it says:

... likes to ... and ... likes .....

Luckily, it was recorded in a positional index. But unfortunately, they also lost some parts of the index and only have the lists for the following terms:

she: **1**: [1, 5], **2**: [1], **3**: [3], **4**: [3], **5**: [1]

eat: **3**: [8], **2**: [4], **5**: [8]

pink: **4**: [8], **5**: [7]

recipe: **3**: [2]

cook: **1**: [4], **5**: [4]

Which document could potentially be document 2?

Note: We assume that the first word in the document is automatically capitalized and that full stop is added at the end of the document.

She likes to eat and he likes to wink.

FS22 – Question 12

Due to a natural catastrophe, some information was lost in the data center in which a library stored its documents and inverted indexes

In particular, they lost some parts of the document with ID 2. They now only know that it says:

... likes to ... and ... likes .....

Luckily, it was recorded in a positional index. But unfortunately, they also lost some parts of the index and only have the lists for the following terms:

she: **1**: [1, 5], **2**: [1], **3**: [3], **4**: [3], **5**: [1]

eat: **3**: [8], **2**: [4], **5**: [8]

pink: **4**: [8], **5**: [7]

recipe: **3**: [2]

cook: **1**: [4], **5**: [4]

Which document could potentially be document 2?

Note: We assume that the first word in the document is automatically capitalized and that full stop is added at the end of the document.

She likes to eat and he likes to wink.

Keyword: potentially
The postings list for wink was not recovered, so there is nothing that goes against this document being in the data center.

FS22 – Question 12

Due to a natural catastrophe, some information was lost in the data center in which a library stored its documents and inverted indexes

In particular, they lost some parts of the document with ID 2. They now only know that it says:

... likes to ... and ... likes .....

Luckily, it was recorded in a positional index. But unfortunately, they also lost some parts of the index and only have the lists for the following terms:

she: **1**: [1, 5], **2**: [1], **3**: [3], **4**: [3], **5**: [1]

eat: **3**: [8], **2**: [4], **5**: [8]

pink: **4**: [8], **5**: [7]

recipe: **3**: [2]

cook: **1**: [4], **5**: [4]

Which document could potentially be document 2?

Note: We assume that the first word in the document is automatically capitalized and that full stop is added at the end of the document.

She likes to eat and he likes the recipe.

FS22 – Question 12

Due to a natural catastrophe, some information was lost in the data center in which a library stored its documents and inverted indexes

In particular, they lost some parts of the document with ID 2. They now only know that it says:

... likes to ... and ... likes .....

Luckily, it was recorded in a positional index. But unfortunately, they also lost some parts of the index and only have the lists for the following terms:

she: **1**: [1, 5], **2**: [1], **3**: [3], **4**: [3], **5**: [1]

eat: **3**: [8], **2**: [4], **5**: [8]

pink: **4**: [8], **5**: [7]

recipe: **3**: [2]

cook: **1**: [4], **5**: [4]

Which document could potentially be document 2?

Note: We assume that the first word in the document is automatically capitalized and that full stop is added at the end of the document.

She likes to eat and he likes the recipe.

The postings list for “recipe” has no posting for document 2.

FS19 – Question 17

For each of the following statements, mark whether it refers to the positional index or the biword index.

Positional index	Biword index	
○	○	This index has a larger number of postings lists than the other.
○	○	The original documents (lists of words) can be fully reconstructed from the index. (For this statement, we consider that no post-processing beyond tokenization was performed to build the index).
○	○	Using this index for phrase search can lead to false positives.
○	○	This index needs more space per term.

FS19 – Question 17

For each of the following statements, mark whether it refers to the positional index or the biword index.

Positional index	Biword index	
○	○	This index has a larger number of postings lists than the other.
○	○	The original documents (lists of words) can be fully reconstructed from the index. (For this statement, we consider that no post-processing beyond tokenization was performed to build the index).
○	○	Using this index for phrase search can lead to false positives.
○	○	This index needs more space per term.

Biword index, since the term vocabulary grows exponentially in k-word indices and will thus have more postings lists.

Positional index, since it saves the exact position in the documents.

Biword index

Positional index, since it also stores the exact positions.