Introduction to
Information Retrieval

02 The term vocabulary & postings lists

2.1 Document delineation and character sequence decoding
2.2 Determining the vocabulary of terms
2.3 Faster postings list intersection via skip pointers

BK인턴 유승미

Contents

2.1 Document delineation and character sequence decoding

2.2 Determining the vocabulary of terms

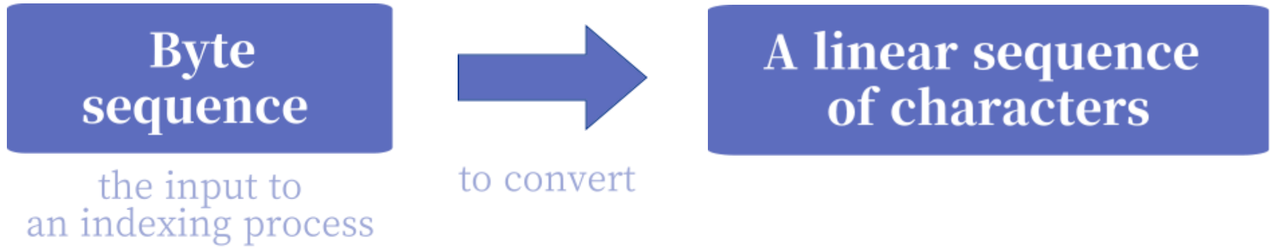
2.3 Faster postings list intersection via skip pointers

The major steps in inverted index construction

1. Collect the documents to be indexed.
2. Tokenize the text.
3. Do linguistic preprocessing of tokens.
4. Index the documents that each term occurs in

1. Obtaining the character sequence in a document

- The first step of processing



1 **Encoding** (two cases)

1. plain English text in ASCII encoding **simple**
2. one of various single byte or multibyte encoding schemes **complex**

   (such as Unicode UTF-8, or various national or vendor-specific standards.)

1. Obtaining the character sequence in a document

2 Decoding (Once the encoding is determined)

- some binary representation like Microsoft Word DOC files
- a compressed format (zip files)
- plain text documents (XML documents)

  (additional decoding for character entities)

2. Choosing a document unit

- The next phase : to determine what the document unit for indexing is.
  - very long documents

    ➡ the issue of indexing *granularity* arise

ex) For a collection of books

- to index an entire book as a document ✕
- to index each chapter or parahraph as a mini-documents ○

➡ more likely to be relevant,
and since the documents are smaller it will be much easier
for the user to find the relevant passages in the document.

2. Choosing a document unit

- We could treat individual sentences as mini-documents.

a precision/recall tradeoff

the units — if

- too small → we are likely to miss important passages
- too large → we tend to get spurious matches and the relevant information is hard for the user to find.

" An IR system should be designed to offer choices of granularity "

Tokenization

- Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces(called tokens).

  ex) Input: Friends, Romans, Countrymen, lend me your ears;
  Output: | Friends | Romans | Countrymen | lend | me | your | ears |

Token an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing.

Type the class of all tokens containing the same character sequence

Term a (perhaps normalized) type that is included in the IR system's dictionary

Tokenization

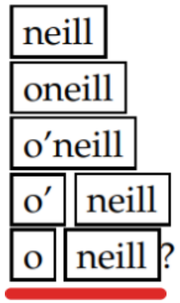
- The major question of the tokenization phase

  " what are the correct tokens to use? "

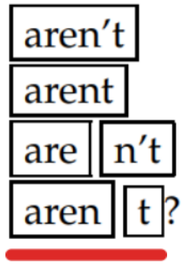
- For example,
  what do you do about the various uses of the apostrophe?

  → split on all non-alphanumeric characters (A simple strategy)

For *O'Neill*, which of the following is the desired tokenization?



And for *aren't*, is it:



Source: https://nlp.stanford.edu/IR-book/information-retrieval-book.html

Tokenization

- These issues of tokenization are language-specific.

French

- uses an apostrophe for a reduced definite article 'the' before a word starting with a vowel. (e.g., l'ensemble)
- uses a hyphen with postposed clitic pronouns in imperatives and questions. (e.g., donnemoi 'give me')

German

- writes compound nouns without spaces. (e.g., Computerlinguistik 'computational linguistics')

Major East Asian languages (e.g., Chinese, Japanese, Korean, and Thai)

- write text without any spaces between words.

Tokenization

Major East Asian languages (e.g., Chinese, Japanese, Korean, and Thai)

- write text without any spaces between words.

莎拉波娃现在居住在美国东南部的佛罗里达。今年４月９日，莎拉波娃在美国第一大城市纽约度过了１８岁生日。生日派对上，莎拉波娃露出了甜美的微笑。

→ abandon word-based indexing

and do all indexing via just short subsequences of characters (character k-grams)

(regardless of whether particular sequences cross word boundaries or not)

Source: https://nlp.stanford.edu/IR-book/information-retrieval-book.html

Dropping common terms: stop words

stop words

some extremely common words which would appear to be of little value in helping select documents matching a user need.

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					

► **Figure 2.5** A stop list of 25 semantically non-selective words which are common in Reuters-RCV1.

- Using a stop list significantly reduces the number of postings that a system has to store.

Source: https://nlp.stanford.edu/IR-book/information-retrieval-book.html

Dropping common terms: stop words

stop words

- Sort the terms by collection frequency
- Take the most frequent terms
- Hand-filter for semantic content relative to the domain
- Create a stop list
- Discard stop list members during indexing

- The general trend ind IR systems

large stop lists (200-300 terms) › **small stop lists** (7-12 terms) › **no stop list**

Normalization (equivalence classing of terms)

Token normalization

the process of canonicalizing tokens so that matches occur despite superficial differences in the character sequences of the tokens.

ex) when searching for "USA," you may also want documents containing "U.S.A." to be included.

Query term	Terms in documents that should be matched
Windows	Windows
windows	Windows, windows, window
window	window, windows

► **Figure 2.6** An example of how asymmetric expansion of query terms can usefully model users' expectations.

Source: https://nlp.stanford.edu/IR-book/information-retrieval-book.html

Normalization (equivalence classing of terms)

- The most standard way to normalize: create equivalence classes

ex) *anti-discriminatory* / *antidiscriminatory* → *antidiscriminatory*

- It is only easy to write rules of this sort that remove characters.
- Since the equivalence classes are implicit,
  it is not obvious when you might want to add characters.

Normalization (equivalence classing of terms)

- An alternative to creating equivalence classes is
  to maintain relations between unnormalized tokens.

- Two ways

1. Query Expansion

   (to maintain a query expansion list of multiple vocabulary entries to consider for a certain query term.)

2. Index Expansion

   (Stores multiple forms of a word during index creation.)

Stemming and lemmatization

- Goal: to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

> am, are, is ⇒ be
> car, cars, car's, cars' ⇒ car
>
> The result of this mapping of text will be something like:
>
> the boy's cars are different colors ⇒
> the boy car be differ color

Stemming

A crude heuristic process that chops off the ends of words, often **removing derivational affixes**.

ex) saw ⇒ s

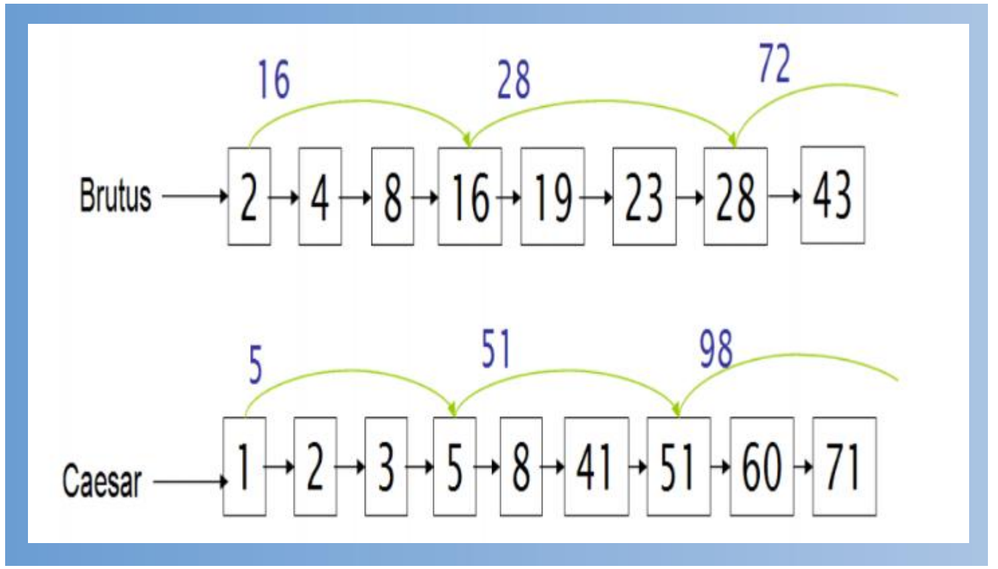
Lemmatization

Uses vocabulary and morphological analysis to properly remove inflectional endings and return the base or dictionary form (lemma) of a word.

ex) saw (a verb) ⇒ see
saw (a noun) ⇒ saw

Source: https://nlp.stanford.edu/IR-book/information-retrieval-book.html

Postings lists with skip pointers




```
INTERSECTWITHSKIPS(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then if hasSkip(p1) and (docID(skip(p1)) ≤ docID(p2))
 9                     then while hasSkip(p1) and (docID(skip(p1)) ≤ docID(p2))
10                          do p1 ← skip(p1)
11                     else p1 ← next(p1)
12                else if hasSkip(p2) and (docID(skip(p2)) ≤ docID(p1))
13                     then while hasSkip(p2) and (docID(skip(p2)) ≤ docID(p1))
14                          do p2 ← skip(p2)
15                     else p2 ← next(p2)
16  return answer
```

Source: https://nlp.stanford.edu/IR-book/information-retrieval-book.html

감사합니다.