Introduction to

Information Retrieval

Term vocabulary and postings lists – preprocessing steps

Documents

- Last lecture: Simple Boolean retrieval system
- Our assumptions were:
  - We know what a document is.
  - We can “machine-read” each document.
- This can be complex in reality.

Parsing a document

- Convert byte sequence into a linear sequence of characters
- Requirements
  - Deal with format and language of each document
  - What is the encoding? E.g., UTF-8
  - What format is it in? pdf, word, excel, html, etc.
  - What language is it in?
  - What character set is in use?
- Each of these is a classification problem
- Alternative: use heuristics

Format/Language: Complications

- A single index usually contains terms of several languages.

- A document may contain multiple languages/formats
  - French email with Spanish pdf attachment
  - Code switching in social media

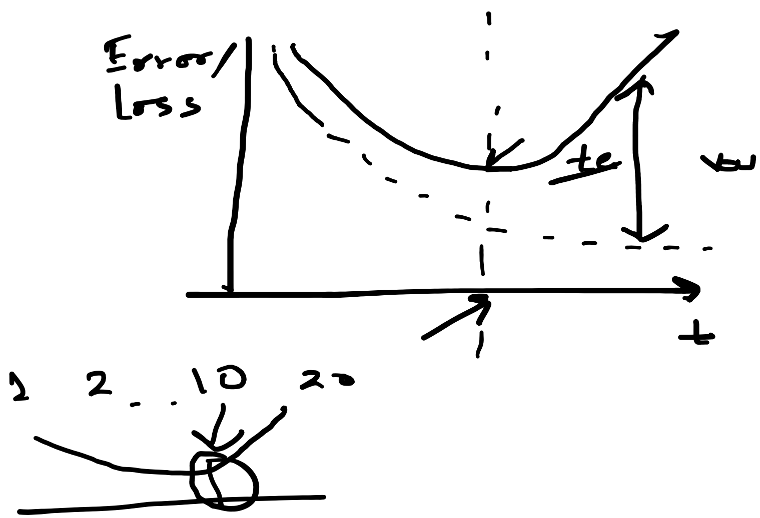
Format/Language: Complications

- What is the document unit for indexing?
  - A file?
  - An email? An email file can contain a sequence of messages; each message can be considered a document
  - An email with 5 attachments?
  - Multiple files may be combined into one document (ppt or latex in HTML)
- Sentence → paragraph → Chapters → books
- Upshot: Answering the question “what is a document?” is not trivial and requires some design decisions.

Train

Valid

Test



Definitions

equiv. rel'n '~' → reflexive, symm., trans

x ~ x

x ~ y y ~ x

x ~ y, y ~ z x ~ z

means-the-same

should-match

kill → killed, killing, kills

murder

killer

- Word – A delimited string of characters as it appears in the text
- Term – A "normalized" word (case, morphology, spelling etc); actually an equivalence class of words; usually what is included in an IR system's dictionary
- Token – An instance of a word or term occurring in a document.
- Type – The same as a term in most cases: an equivalence class of tokens.

a sleep perchance to dream

token = 5 term = 3

Recall: Inverted index construction

- Input:

Friends, Romans, countrymen. | So let it be with Caesar | . . .

- Output:

friend | roman | countryman | so | . . .

- Each token is a candidate for a postings entry.
- What are valid tokens to emit?

Exercises

In June, the dog likes to chase the cat in the barn.

– How many word tokens? How many word types?

Why tokenization is difficult even in English?

Tokenize: *Mr. O'Neill thinks that the boys' stories about Chile's capital aren't amusing.*

o neill

oneill

neill

o 'neill

AND

$\frac{1}{2}$

$\frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 0 + \frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 0$

Tokenization problems: One word or two? (or several)

- 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 problems: Numbers

- 3/20/91
- 20/3/91
- Mar 20, 1991
- B-52 (aircraft)
- 100.2.86.144
- (800) 234-2333
- 800.234.2333
- Older IR systems may not index numbers . . .
- . . . but generally it’s a useful feature.

Problems in tokenization for other languages, e.g., no whitespace in Chinese

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。

Ambiguous segmentation in Chinese

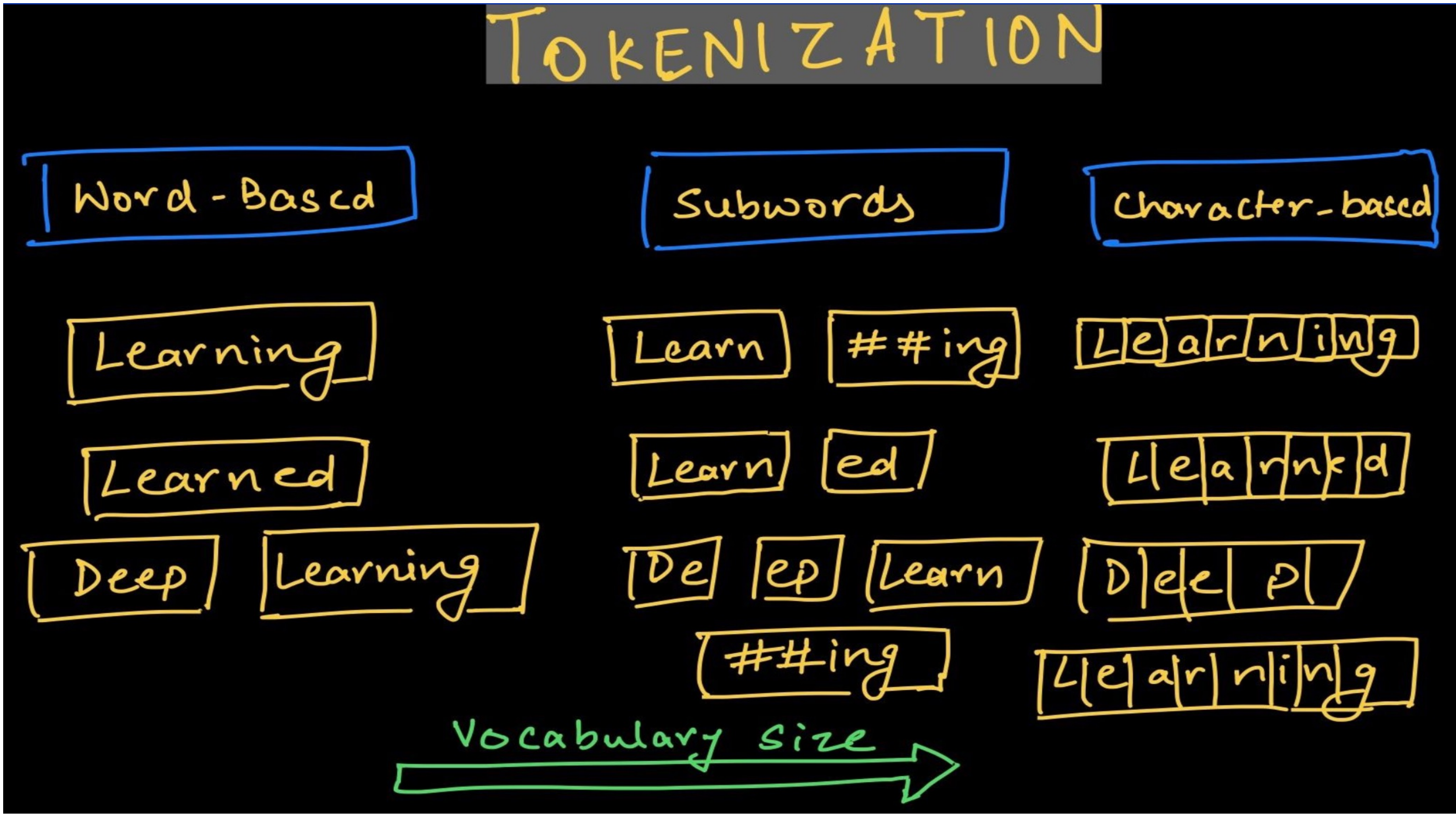
和尚

The two characters can be treated as one word meaning 'monk' or as a sequence of two words meaning 'and' and 'still'.

Other cases of “no whitespace”

- Compounds in Dutch, German, Swedish
- Computerlinguistik → Computer + Linguistik
- Lebensversicherungsgesellschaftsangestellter
- → leben + versicherung + gesellschaft + angestellter
- Inuit: tusaatsiarunnanngittualuujunga (I can’t hear very well.)
- Many other languages with segmentation difficulties: Finnish, Urdu, . . .
- कमला-कुच-कुङ्कुम-पिञ्जरीकृत-वक्षः-स्थल-विराजित-महा-कौस्तुभ-मणि-मरीचि-माला-निराकृत-त्रि-भुवन-तिमिर
- CamelCase in social media

Evolution of Tokenization Strategies



Hidden Markov Model Basics

Task: Word Segmentation (similar to NP chunking)

- Input: c o m p . l i n g
- Output: B I I I O B I I I

Generative modeling: what is the generative probability of the character sequence $p(C|\theta)$? where $C = c_1, c_2, \ldots c_n$, $T = t_1, t_2, \ldots, t_n$

$$p(C|\theta) = \sum_T p(C|T)p(T|\theta)$$

$$= \sum_T \Pi_i \underbrace{p(c_i|t_i)}_{\text{Emission Probability}} \overbrace{p(t_i|t_{i-1}, \theta)}^{\text{Transition Probability}}$$

Inference (MAP): $T = \max_T p(T|C) \approx \max_T \Pi_i\, p(c_i|t_i)p(t_i|t_{i-1})$

HMM Basics

NN NN

<s> comp . Ling </s>

<s> B I I I O B I I I </s>

$$P(w_1 \dots w_n) = \sum_{t_1 \dots t_n} P(w_1 \dots w_n \mid t_1 \dots t_n)\, P(t_1 \dots t_n)$$

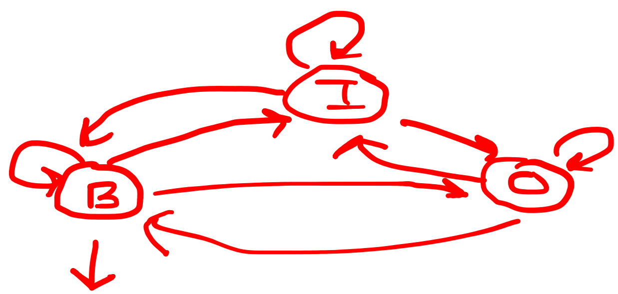
$$= \sum \prod_i P(w_i \mid t_i) \quad B$$

$$\prod_i P(t_i \mid t_{i-1}) \quad A$$

$$\sum_i P(s_i \mid B) = 1$$

I

O



$P(c \mid B)$

$P(o \mid B)$

$\vdots$

$P(g \mid B)$

$$P(T \mid W)$$

$$\hat{T} = \arg\max_T P(T \mid W; \theta)$$

Arabic script

ك ِ ت ا ب ٌ ⇐ كِتابٌ

un b ā t i k

/kitābun/ *'a book'*

peña = cliff

pena sorrow

Arabic script: Bidirectionality

استقلت الجزائر في سنة 1962 بعد 132 عاما من الاحتلال الفرنسي.

← → ← → ← START

'Algeria achieved its independence in 1962 after 132 years of French occupation.'

Bidirectionality is not a problem if text is coded in Unicode.

Accents and diacritics

- Accents: résumé vs. resume (simple omission of accent)
- Umlauts: Universität vs. Universitaet (substitution with special letter sequence "ae")
- Most important criterion: How are users likely to write their queries for these words?

  peña penlua

- Even in languages that standardly have accents, users often do not type them. (Polish?)

Case folding

Julius → julius

I → I

- Usually: Reduce all letters to lower case
- Possible exceptions: capitalized words in mid-sentence
- MIT vs. mit

CompLing

- Fed vs. fed
- It's often best to lowercase everything since users will use lowercase regardless of correct capitalization

Normalization

- Need to "normalize" terms in indexed text as well as query terms into the same form.
- Example: We want to match *U.S.A.* and *USA*
- We commonly implicitly define equivalence classes of terms
  - Can use hand-constructed rules, e.g., 'car' & 'automobile'
- Alternatively: do asymmetric expansion
  - window → window, windows
  - windows → Windows, windows
  - os → Windows (no expansion)
- More powerful, but less efficient
- Why don't you want to put *window, Windows, windows,* and *Windows* in the same equivalence class?

cars → car
syn(car) → car
jaguar → car

Normalization: Other languages

- Normalization and language detection interact.
- *PETER WILL NICHT MIT.* → MIT = mit
- *He got his PhD from MIT.* → MIT ≠ mit

Stop words

- stop words: extremely common words which would appear to be of little value in helping select documents matching a user need
- Examples: *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*
- Stop word elimination used to be standard in older IR systems.
- But you need stop words for phrase queries, e.g. "King of Denmark"
- Most web search engines index stop words.

Stemming and Lemmatization

- Goal of both same: reduce inflectional forms and derivationally related forms to a common base form
- Stemming refers to a heuristic process that chops off the ends of words in the hope of achieving the goal correctly most of the time
- Lemmatization implies doing “proper” reduction to dictionary headword form (the lemma), using dictionary and morphological analysis of words

Lemmatization

- Reduce inflectional/variant forms to base form
- Example: *am, are, is* → *be*
- Example: *car, cars, car's, cars'* → *car*
- Example: *the boy's cars are different colors* → *the boy car be different color*
- Inflectional morphology (*cutting* → *cut)* vs. derivational morphology (*destruction* → *destroy)*

Stemming

- Heuristic process that chops off the ends of words in the hope of achieving what "principled" lemmatization attempts to do with a lot of linguistic knowledge.
- Language dependent
- Often inflectional and derivational
- Example for derivational: *automate, automatic, automation* all reduce to *automat*

Porter algorithm

- Most common algorithm for stemming English
- Results suggest that it is at least as good as other stemming options
- Conventions + 5 phases of reductions
- Phases are applied sequentially
- Each phase consists of a set of commands.
  - Sample command: Delete final *ement* if what remains is longer than 1 character
  - replacement → replac
  - cement → cement
- Sample convention: Of the rules in a compound command, select the one that applies to the longest suffix.

Porter stemmer: A few rules

Rule	**Example**
SSES → SS	caresses → caress
IES → I	ponies → poni
SS → SS	caress → caress
S →	cats → cat

Three stemmers: A comparison

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

Lovins stemmer: 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 stemmer: 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

Does stemming improve effectiveness?

- In general, stemming increases effectiveness for some queries, and decreases effectiveness for others.
- Queries where stemming is likely to help: [tartan sweaters], [sightseeing tour san francisco] (equivalence classes: {sweater,sweaters}, {tour,tours})
- Porter Stemmer equivalence class *oper* contains all of *operate operating operates operation operative operatives operational.*
- Queries where stemming hurts: [operational AND research], [operating AND system], [operative AND dentistry]