CMPE 493
INTRODUCTION TO
INFORMATION RETRIEVAL

Basic Text Preprocessing

Department of Computer Engineering, Boğaziçi University
November 2, 2020

Plan for this lecture

Elaborate basic indexing

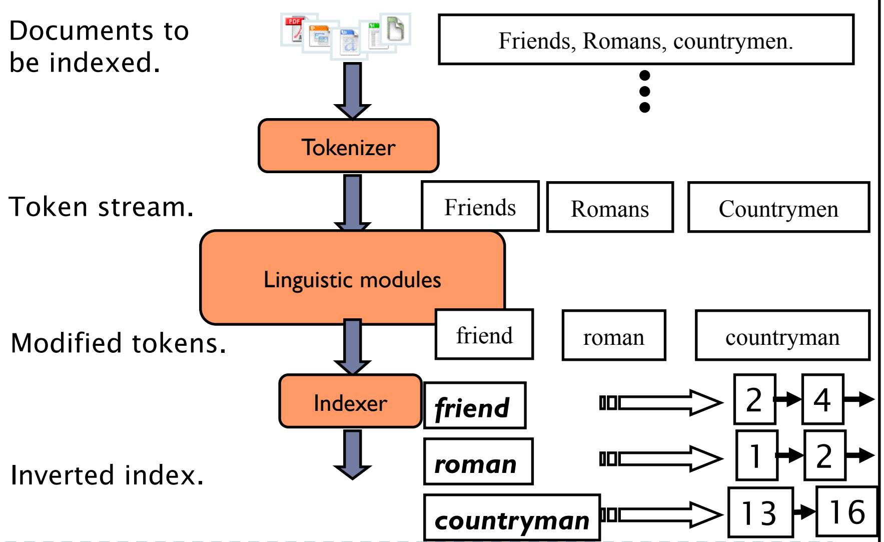
- Preprocessing to form the term vocabulary
  - Documents
  - Tokenization
  - What *terms* do we put in the index?

Exercise

- Draw the inverted index that would be built for the following document collection.

Doc 1	new home sales top forecasts
Doc 2	home sales rise in july
Doc 3	increase in home sales in july
Doc 4	july new home sales rise

Recall the basic indexing pipeline



Parsing a document

- What format is it in?
  - pdf/word/excel/html?
- What language is it in?
- What character set is in use?
  - ASCII, Unicode UTF-8, national or vendor specific

Each of these is a classification problem, which we will study later in the course.

But these tasks are often done heuristically …

Complications: Format/language

- Documents being indexed can include docs from many different languages
  - A single index may have to contain terms of several languages.
- Sometimes a document or its components can contain multiple languages/formats
  - Turkish email with an English pdf attachment.
- What is a unit document?
  - A file?
  - An email?
  - An email with 5 attachments?
  - A group of files (PPT or LaTeX as HTML pages)

Tokens and Terms

Tokenization

- Input: "***Friends, Romans and Countrymen***"
- Output: Tokens
  - ***Friends***
  - ***Romans***
  - ***and***
  - ***Countrymen***
- A token is an instance of a sequence of characters
- Each such token is now a candidate for an index entry, after further processing
  - Described below
- But what are valid tokens to emit?

Tokenizing

- For English, why not just use white-space?
  - `Mr. Sherwood said reaction to Sea Containers' proposal has been "very positive." In New York Stock Exchange composite trading yesterday, Sea Containers closed at $62.625, up 62.5 cents.`
  - `"I said, 'what're you? Crazy?' " said Sadowsky. "I can't afford to do that.''`
- Using white-space gives you words like:
  - `cents.`
  - `said,`
  - `positive."`
  - `Crazy?'`

Tokenization

- Issues in tokenization:
  - ***Turkey's capital*** →
    Turkey? Turkeys? Turkey's?
  - **Clitics:** We're, I'am, I've, isn't
  - ***Hewlett-Packard*** → ***Hewlett*** and ***Packard*** as two tokens?
    - ***state-of-the-art***: break up hyphenated sequence?
    - ***co-education, coeducation***
    - ***lowercase***, ***lower-case***, ***lower case*** ?
    - **New York-San Francisco flight**
    - **Ph.D.**
    - **AT&T**

Numbers

- ***2/11/2020 Feb. 11, 2020 11/2/2020***
- ***2/11/2011 2 November 2011***
- ***55 B.C.***
- ***(800) 234-2333***

  - Often have embedded spaces
  - Older IR systems may not index numbers
    - But often very useful: think about things like looking up error codes/stacktraces on the web
  - Will often index "meta-data" separately
    - Creation date, format, etc.

Tokenization: language issues

- French
  - ***L'ensemble*** → one token or two?
    - ***L*** ? ***L'*** ? ***Le*** ?
    - Want ***l'ensemble*** to match with ***un ensemble***
- German noun compounds are not segmented
  - ***Lebensversicherungsgesellschaftsangestellter***
  - 'life insurance company employee'
  - German retrieval systems benefit greatly from a **compound splitter** module

Sec. 2.2.1

Tokenization: language issues

- Chinese and Japanese have no spaces between words:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - Not always guaranteed a unique tokenization
- Further complicated in Japanese, with multiple alphabets intermingled
  - Dates/amounts in multiple formats

フォーチュン500社は情報不足のため時間あた$500K(約6,000万円)

Katakana Hiragana Kanji Romaji

End-user can express query entirely in Hiragana!

Maximum Matching Word Segmentation

Baseline greedy algorithm.

Given a lexicon of Chinese, and a string

1) Start a pointer at the beginning of the string
2) Find the longest word in dictionary that matches the string starting at pointer
    1) If there are no matches, emit a character and advance the pointer 1 character
3) Move the pointer over the word in string
4) Go to 2

Max-match segmentation

- thecatinthehat
  - the cat in the hat
- thetabledownthere
  - the table down there
  - theta bled own there

Dictionary:
bled
cat
down
hat
in
own
table
the
there
theta

- Doesn’t really work for English.
- Often pretty successful for Chinese.
- ML methods more successful (Hidden Markov Models, Conditional Random Fields, Recurrent Neural Networks etc.)

Practical English Segmentation Examples

- URL segmentation
  - www.dietsthatwork.com
  - www.choosespain.com
- Twitter hashtag segmentation
  - #unitedbrokemyguitar
  - #manchesterunited

Hashtag Segmentation

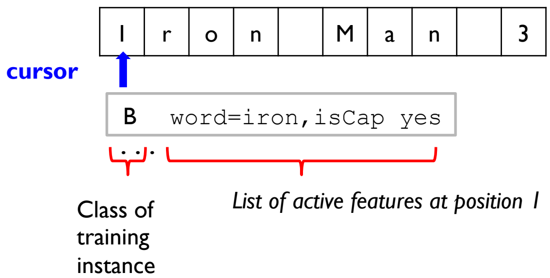
- Phrases, partial or full sentences turned into hashtags
- Complex numeric expressions
- Not much help of capitalization
- Words with sentiment trapped inside hashtags
- Named entities as hashtags

Arda Çelebi and Arzucan Özgür. Segmenting hashtags and analyzing their grammatical structure. Journal of the Association for Information Science and Technology (JASIST). doi:10.1002/asi.23989, 2018.

Feature-based Approach

- Each character in input converted to a training instance

  e.g. "Iron Man 3"



Context Features

Tweet 1:

Wine lovers, we think you'll understand :) #nationalwineday

word-in-context=4+3

orthographic-shape-distribution-in-context=cCc-ccc

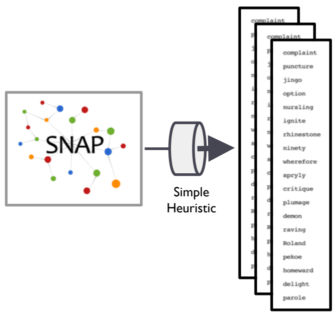
Tweet 2:

#TwitterisblockedinTurkey It's unacceptable to prevent one from expressing oneself!!!

most-sem-similar-word-in-context=0.7

Training Set Creation

Using raw text (i.e., tweets)
w/ existing word boundaries



Using ~1M segmented hashtags (***H***),
auto-extracted w/ simple heuristic

Tokenization: language issues

- Arabic (or Hebrew) is basically written right to left, but with certain items like numbers written left to right
- Words are separated, but letter forms within a word form complex ligatures

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

← start

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

Stop words

- With a stop list, you exclude from the dictionary the commonest words. Intuition:
  - They have little semantic content: *the, a, and, to, be*
  - There are a lot of them.
- But the trend is away from doing this:
  - Good compression techniques (will be discussed later in the course) means the space for including stopwords in a system is very small
  - Good query optimization techniques mean you pay little at query time for including stop words.
  - You need them for:
    - Phrase queries: "King of Denmark"
    - Various song titles, etc.: "Let it be", well known verse "To be or not to be"
    - "Relational" queries: "flights to İstanbul"

Normalization to terms

- We need to "normalize" words in indexed text as well as query words into the same form
  - We want to match ***U.S.A.*** and ***USA***
- Result is terms: a term is a (normalized) word type, which is an entry in our IR system dictionary
- We most commonly implicitly define equivalence classes of terms by, e.g.,
  - deleting periods to form a term
    - ***U.S.A., USA → USA***
  - deleting hyphens to form a term
    - ***anti-discriminatory, antidiscriminatory → antidiscriminatory***

Normalization: other languages

- Accents: e.g., French ***résumé*** vs. ***resume.***
- Umlauts: e.g.,
  - German: ***Tuebingen*** vs. ***Tübingen***
  - Turkish: ***Bogazici vs. Boğaziçi***
  - Should be equivalent
- Most important criterion:
  - How are your users likely to write their queries for these words?
- Even in languages that standardly have accents, users often may not type them
  - Often best to normalize to a de-accented term
    - ***Tuebingen, Tübingen, Tubingen → Tubingen***
- ***Very challenging for some domains such as social media text***
  - ***u -> you; ur -> your; be4 -> before; u 2 -> you too***

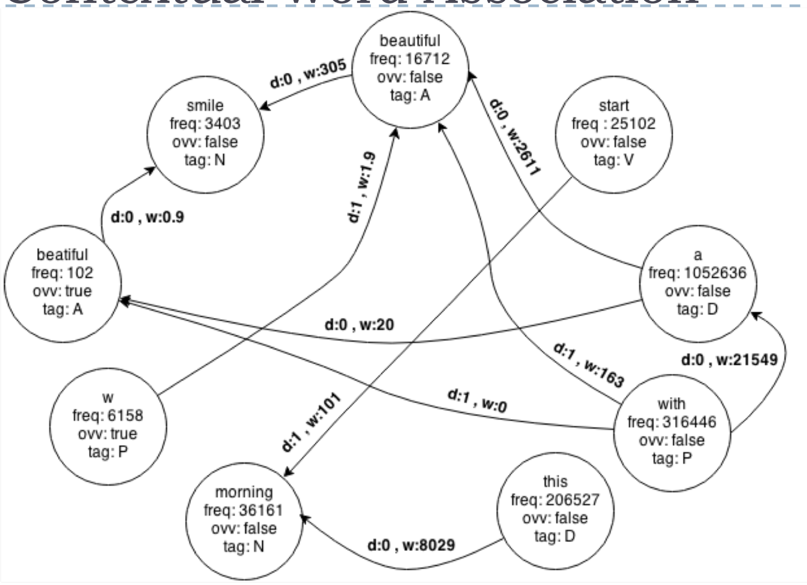
Social Text Normalization

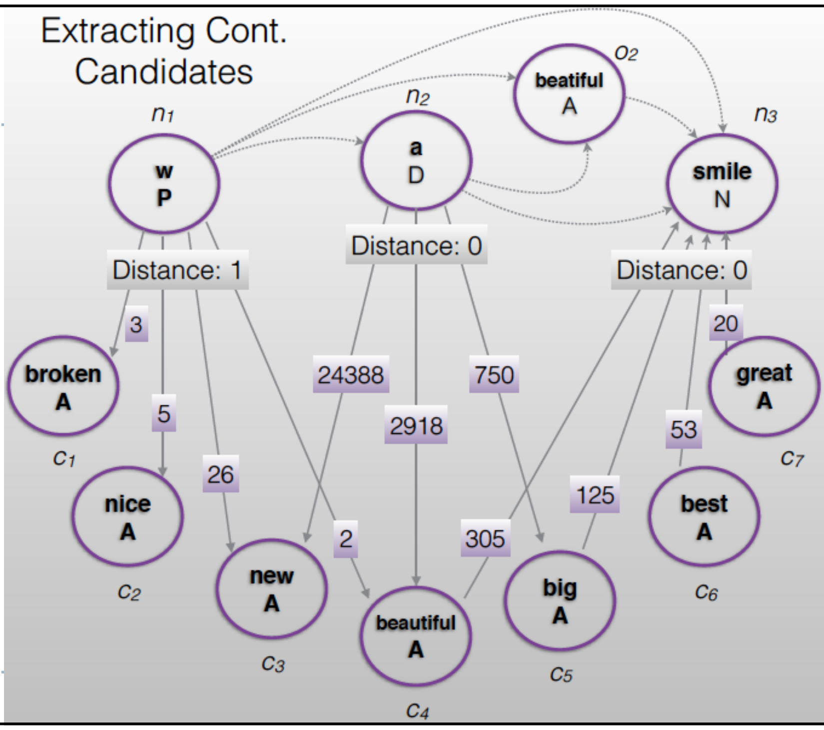
Its a btf nite, lukin for smth fun to do,
I think I wanna be w ma frnds.

Its a beautiful night, looking for something fun to do,
I think I want to be with my friends.

- Çağil Sönmez and Arzucan Özgür. A Graph-based Approach for Contextual Text Normalization. Proceedings of the EMNLP, 2014.

Contextual Word Association





Case folding

- Reduce all letters to lower case
  - exception: upper case in mid-sentence?
    - e.g., ***General Motors***
    - ***Fed*** vs. ***fed***
  - Often best to lower case everything, since users will use lowercase regardless of 'correct' capitalization…
- Example:
  - Query ***C.A.T.***
  - #1 result from Google used to be for "cat" (animal) *not* Caterpillar Inc.
  - Currently, it is Caterpillar Inc.

Normalization: other languages

- Normalization of things like date forms
  - ***7月30日 vs. 7/30***

- Tokenization and normalization may depend on the language and so is intertwined with language detection
- Crucial: Need to "normalize" indexed text as well as query terms into the same form

Morgen will ich in MIT …

Is this German "mit"?

Ich trinke Kaffee **mit** Milch.

Normalization to terms

- An alternative to equivalence classing is to do asymmetric expansion
- An example of where this may be useful
  - Enter: ***window*** Search: ***window, windows***
  - Enter: ***windows*** Search: ***Windows, windows, window***
  - Enter: ***Windows*** Search: ***Windows***
- Potentially more powerful, but less efficient

Thesauri and soundex

- Do we handle synonyms and homonyms?
  - E.g., by hand-constructed equivalence classes, WordNet
    - ***car*** = ***automobile color*** = ***colour***
    - ***jaguar (car) vs. jaguar (animal)***
  - We can rewrite to form equivalence-class terms
    - When the document contains ***automobile***, index it under ***car-automobile*** (and vice-versa)
  - Or we can expand a query
    - When the query contains ***automobile***, look under ***car*** as well
- What about spelling mistakes?
  - One approach is soundex, which forms equivalence classes of words based on phonetic heuristics
  - e.g. Chaikofski should match Tchaikovsky

Lemmatization

- Reduce inflectional/variant forms to base form
- E.g.,
  - *am, are, is* → *be*
  - *car, cars, car's, cars'* → *car*
- *the boy's cars are different colors* → *the boy car be different color*
- Lemmatization implies doing "proper" reduction to dictionary headword form
  - Need a list of grammatical rules + a list of irregular words
  - Children → child, spoken → speak …

Stemming

- Reduce terms to their "roots" before indexing
- "Stemming" suggest crude affix (prefix and suffix) chopping in the hope of achieving what "principled" lemmatization attempts to do with a lot of linguistic knowledge.
  - language dependent
  - e.g., ***automate(s), automatic, automation*** all reduced to ***automat***.

for example compressed and compression are both accepted as equivalent to compress.	→	for exampl compress and compress ar both accept as equival to compress

Porter algorithm

- Most common algorithm for stemming English
- Results suggest that it is at least as good as other stemming options
- 5 phases of reductions + Conventions
- 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 $\rightarrow$ SS	caresses $\rightarrow$ caress
IES $\rightarrow$ I	ponies $\rightarrow$ poni
SS $\rightarrow$ SS	caress $\rightarrow$ caress
S $\rightarrow \emptyset$	cats $\rightarrow$ cat
ING $\rightarrow \emptyset$, if stem has vowel	reading $\rightarrow$ read

Porter Stemmer

- Simple procedure for removing known affixes in English without using a dictionary.
- Can produce unusual stems that are not English words:
  - "computer", "computational", "computation" all reduced to same token "comput"
- May conflate (reduce to the same token) words that are actually distinct.
  - news -> new
  - pretended -> tend
  - glasses -> glass
  - Mrs -> Mr
  - Easter -> East

Other stemmers

- Other stemmers exist, e.g., Lovins stemmer
  - http://www.comp.lancs.ac.uk/computing/research/stemming/general/lovins.htm
  - Single-pass, longest suffix removal (about 250 rules)
- Full morphological analysis – at most modest benefits for retrieval
- Do stemming and other normalizations help?
  - English: very mixed results. Helps recall but harms precision
    - operative (dentistry) ⇒ oper
    - operational (research) ⇒ oper
    - operating (systems) ⇒ oper
  - Useful for Spanish, German, Finnish, …

37

- In English words have no more than 4 or 5 affixes
  - *rewrites*
  - *unbelievably*
- In Turkish words can have up to 9 or 10 affixes

  uygarlastiramadiklarimizdanmissinizcasina

  (behaving) as if you are among those whom we could not civilize

  uygar `civilized' + las `become'
  + tir `cause' + ama `not able'
  + dik `past' + lar 'plural'
  + imiz 'p1pl' + dan 'abl'
  + mis 'past' + siniz '2pl' + casina 'as if'

Dictionary entries

ensemble.french
時間.japanese
MIT.english
mit.german
guaranteed.english
entries.english
sometimes.english
tokenization.english

These may be grouped by language (or not…).

Exercise: What does your favorite search engine do?

- Stop words
- Normalization
- Tokenization
- Lowercasing
- Stemming
- Non-latin alphabets
- Umlauts
- Compounds
- Numbers

References

- *Introduction to Information Retrieval*, chapter 2
  - http://nlp.stanford.edu/IR-book/information-retrieval-book.html
- Porter's stemmer: http://www.tartarus.org/~martin/PorterStemmer/
- Stemming algorithms for various languages including Turkish: https://snowballstem.org
- NLTK
  - http://www.nltk.org
- Zemberek Library for Turkish
  - https://github.com/ahmetaa/zemberek-nlp
- İTÜ Turkish NLP Pipeline for Turkish text pre-processing
  - http://tools.nlp.itu.edu.tr
- Information retrieval on Turkish Texts
  - https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.20750