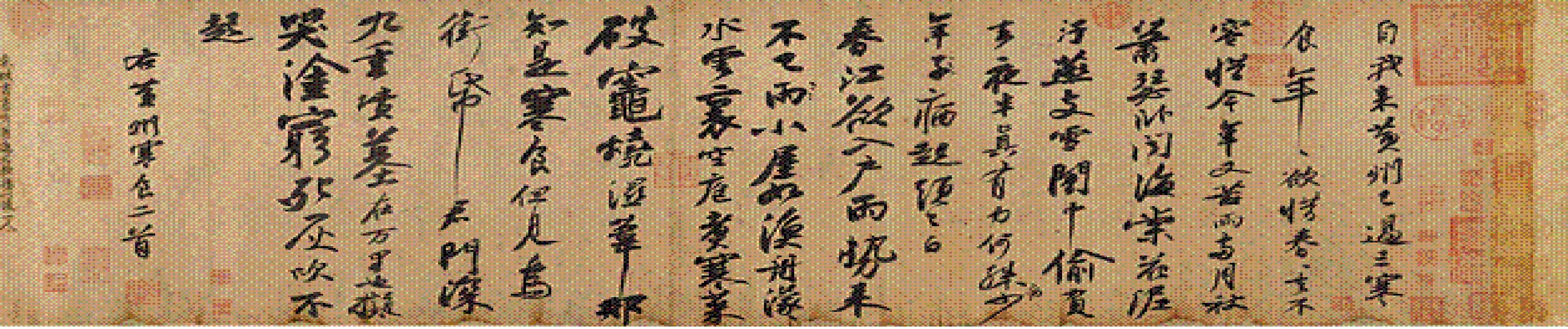


信息检索
Information Retrieval

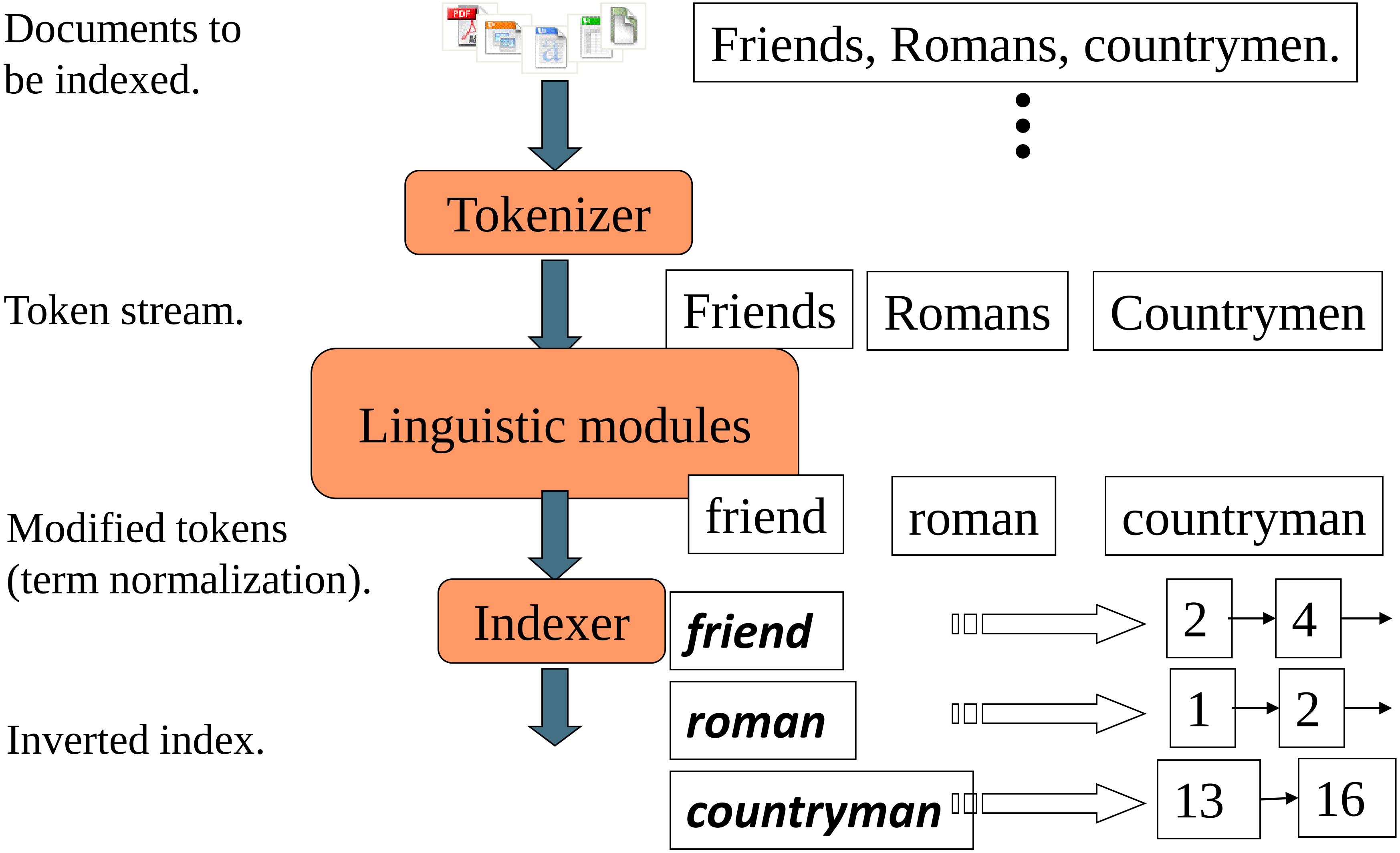
教师：孙茂松
Tel:62781286
Email:sms@tsinghua.edu.cn
TA：**胡锦毅**
Email:hu-jy21@mails.tsinghua.edu.cn

郑重声明

- 此课件仅供选修清华大学计算机系本科生课《信息检索》(40240372)的学生个人学习使用，所以只允许学生将其下载、存贮在自己的电脑中。未经孙茂松本人同意，任何人不得以任何方式扩散之。否则，由此可能引起的一切涉及知识产权的法律责任，概由该人负责。
- 此课件仅限孙茂松本人讲课使用。除孙茂松本人外，凡授课过程中，PTT文件显示此《郑重声明》之情形，即为侵权使用。

第二章 信息检索系统的基本框架（Part 2）

2.3 针对倒排文件的基本操作



Documents to be indexed.

Friends, Romans, countrymen.

Tokenizer

Token stream.

Friends | Romans | Countrymen

Linguistic modules

Modified tokens (term normalization).

friend | roman | countryman

Indexer

Inverted index.

friend → 2 → 4 →

roman → 1 → 2 →

countryman → 13 → 16

Parsing a document

- 基本要求:对文本内容的处理无死角
- What format is it in?
  - pdf/word/excel/html?
- What language is it in?
- What character set is in use?

Each of these is a classification problem.

Complications: Format/language

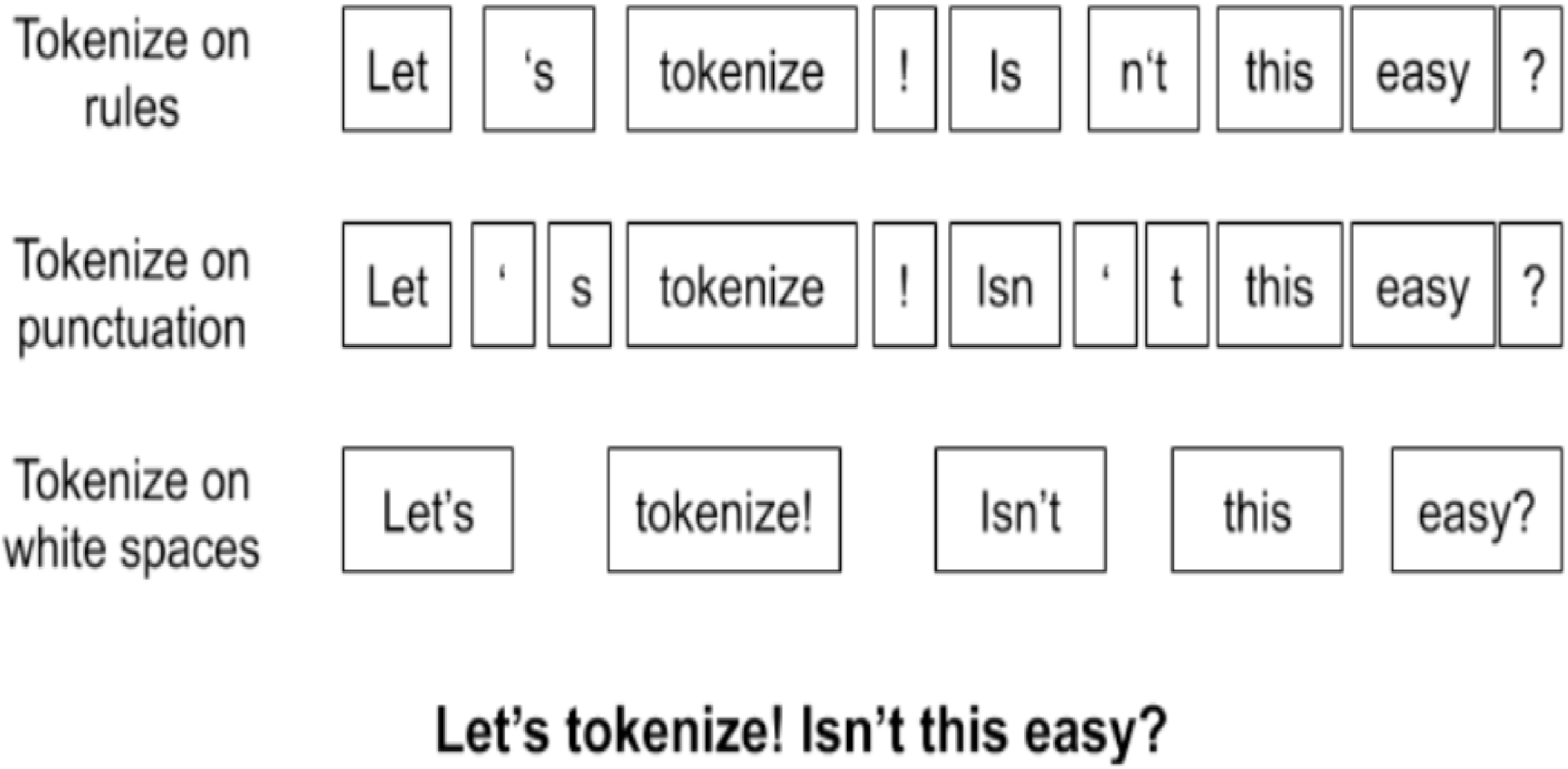
- Documents being indexed can include docs from many different languages
  - A single index may have to contain terms of several languages.
- Sometimes a document or its components can contain multiple languages/formats
  - French email with a German pdf attachment.
- What is a unit document?
  - A file?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

Tokenization

- Input: "***Friends, Romans and Countrymen***"
- Output: Tokens
  - ***Friends***
  - ***Romans***
  - ***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
- Words, Tokens and Terms?

Tokenization



Tokenize on rules: Let | ‘s | tokenize | ! | Is | n‘t | this | easy | ?

Tokenize on punctuation: Let | ‘ | s | tokenize | ! | Isn | ‘ | t | this | easy | ?

Tokenize on white spaces: Let’s | tokenize! | Isn’t | this | easy?

Let’s tokenize! Isn’t this easy?

Tokenization

- Issues in tokenization:
  - ***Finland's capital*** →

    Finland 's? Finland's?
  - ***Hewlett-Packard*** → ***Hewlett*** and ***Packard*** as two tokens?
    - ***state-of-the-art***: break up hyphenated sequence.
    - ***co-education***
    - ***lowercase***, ***lower-case***, ***lower case*** ?
    - It can be effective to get the user to put in possible hyphens
  - ***San Francisco***: one token or two?
    - How do you decide it is one token?

Numbers

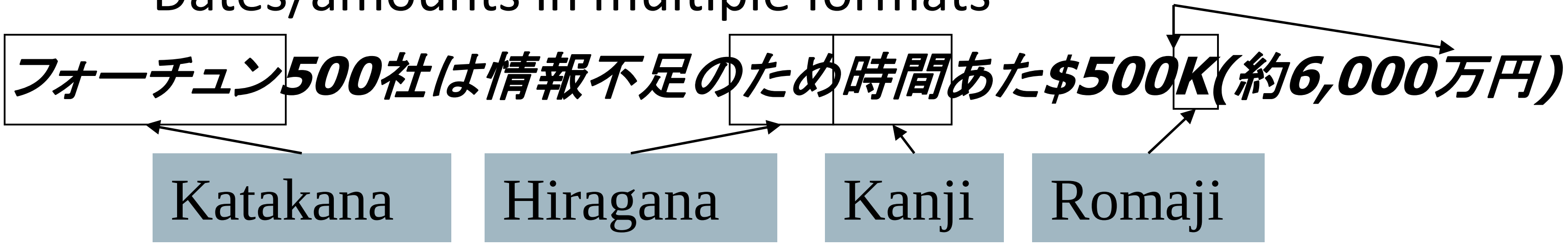
- ***3/20/91*** ***Mar. 12, 1991*** ***20/3/91***
- ***55 B.C.***
- ***B-52***
- ***My PGP key is 324a3df234cb23e***
- ***(800) 234-2333***
  - Often have embedded spaces
  - Older IR systems may not index numbers
    - But often very useful: think about things like looking up error codes/stacktraces on the web
    - (One answer is using n-grams)

Tokenization: language issues

- French
  - ***L'ensemble*** → one token or two?
    - ***L*** ? ***L'*** ?
    - Want ***l'ensemble*** to match with ***un ensemble***
- German noun compounds are not segmented
  - ***Abwasserbehandlungsanlange***
  - *Sewage water treatment plant*
  - ***Abwasser | behandlungs | anlange***
  - German retrieval systems benefit greatly from a **compound splitter** module (Can give a 15% performance boost for German)

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(平假名)!

中文：甲A

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 عاما 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 entirely the commonest words. Intuition:
  - They have little semantic content: *the, a, and, to, be*
  - There are a lot of them: ~30% of postings for top 30 words
- But the trend is away from doing this:
  - Good compression techniques 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”, “To be or not to be”
    - “Relational” queries: “flights to London”

A
ABOUT
ACROSS
AFTER
AFTERWARDS
AGAIN
AGAINST
ALL
ALMOST
ALONE
ALONG
ALREADY
ALSO
ALTHOUGH
ALWAYS
AMONG
AMONGST
AN
AND
ANOTHER
ANY
ANYHOW
ANYONE
ANYTHING
ANYWHERE
ARE
AROUND
AS
AT
BE
BECAME
BECAUSE
BECOME
BECOMES
BECOMING
BEEN
BEFORE
BEFOREHAND
BEHIND
BEING
BELOW
BESIDE
BESIDES
BETWEEN
BEYOND
BOTH
BUT
BY
CAN

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)***

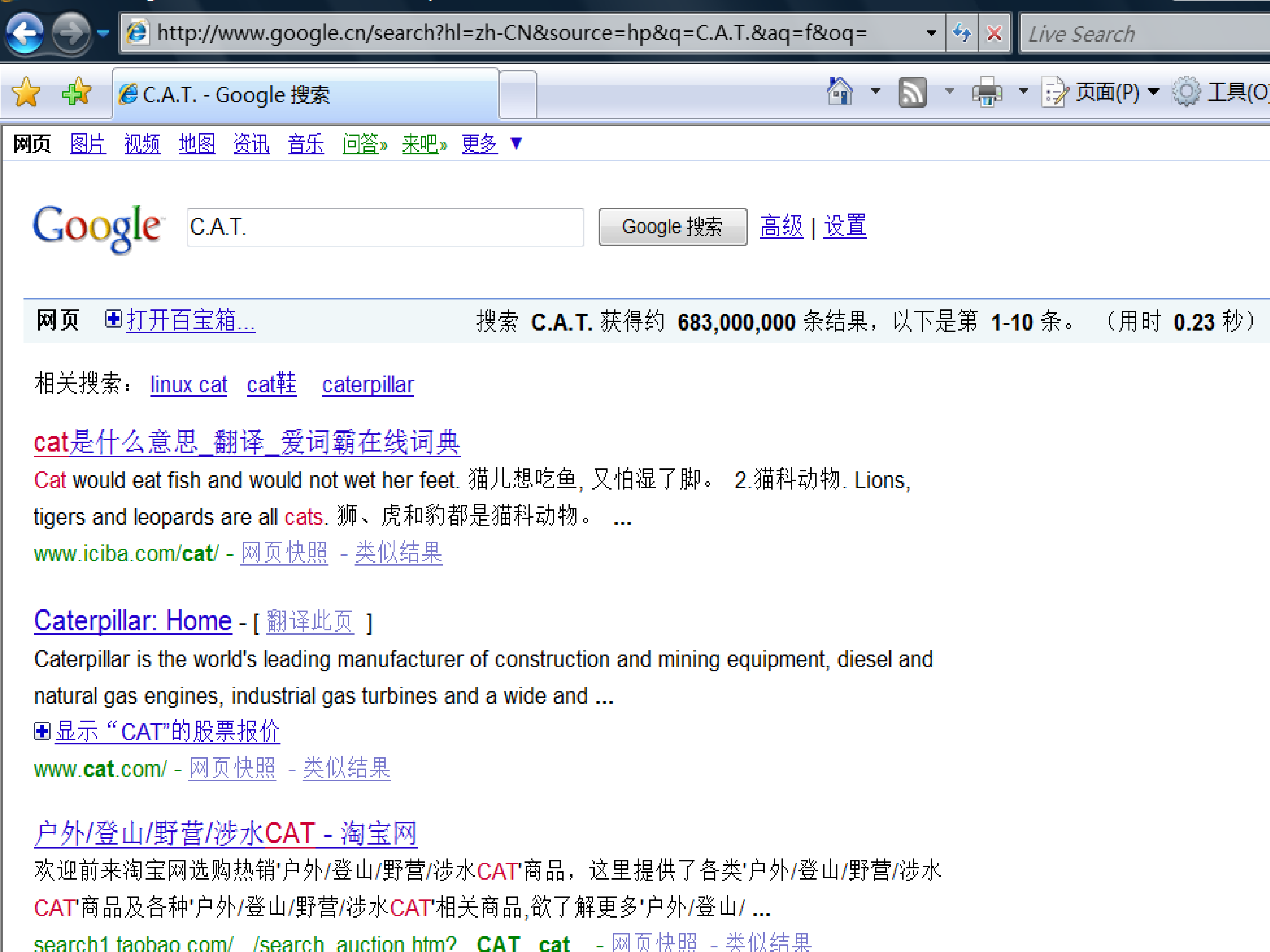
Normalization: other languages

- Accents: e.g., French ***résumé*** vs. ***resume*.**
- Umlauts: e.g., German: ***Tuebingen*** vs. ***Tübingen***
  - Should be equivalent
- Most important criterion:
  - How are your users like 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***

Case folding

- Reduce all letters to lower case
  - exception: upper case in mid-sentence?
    - e.g., ***General Motors***
    - ***SAIL*** vs. ***sail***
  - Often best to lower case everything, since users will use lowercase regardless of ‘correct’ capitalization…
- Google example:
  - Query ***C.A.T.***
  - #1 result is for “cat”, *not* Caterpillar Inc.



搜索 **C.A.T.** 获得约 **683,000,000** 条结果，以下是第 **1-10** 条。 （用时 **0.23** 秒）

相关搜索： linux cat cat鞋 caterpillar

cat是什么意思_翻译_爱词霸在线词典

Cat would eat fish and would not wet her feet. 猫儿想吃鱼，又怕湿了脚。 2.猫科动物. Lions, tigers and leopards are all cats. 狮、虎和豹都是猫科动物。 ...

www.iciba.com/**cat**/ - 网页快照 - 类似结果

Caterpillar: Home - [翻译此页]

Caterpillar is the world's leading manufacturer of construction and mining equipment, diesel and natural gas engines, industrial gas turbines and a wide and ...

显示 “CAT”的股票报价

www.**cat**.com/ - 网页快照 - 类似结果

户外/登山/野营/涉水CAT - 淘宝网

欢迎前来淘宝网选购热销'户外/登山/野营/涉水CAT'商品，这里提供了各类'户外/登山/野营/涉水CAT'商品及各种'户外/登山/野营/涉水CAT'相关商品,欲了解更多'户外/登山/ ...

search1.taobao.com/.../search_auction.htm?...**CAT**...**cat**... - 网页快照 - 类似结果

Lemmatization

- Reduce inflectional/variant forms to base form
- E.g.,
  - *am, are, is* → *be*
  - *car, cars* → *car*
  - *walked, walks or walking* → *walk*
- *the boy's cars are different colors* → *the boy car be different color*
- Lemmatization implies doing "proper" reduction to dictionary headword form

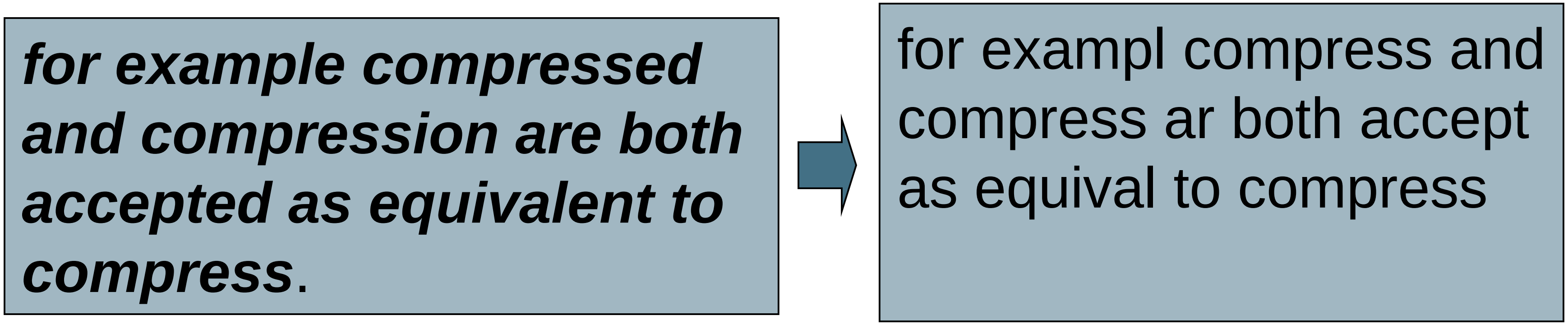
参考 https://stanfordnlp.github.io/CoreNLP/lemma.html

English inflectional affixes, adapted from O'Grady *et al.* 2010:132

Affix	Syntactic/semantic effect	Examples
-s	NUMBER: plural	*cats*
-'s	possessive	*cat's*
-s	TENSE: present, SUBJ: 3sg	*jumps*
-ed	TENSE: past	*jumped*
-ed/-en	ASPECT: perfective	*eaten*
-ing	ASPECT: progressive	*jumping*
-er	comparative	*smaller*
-est	superlative	*smallest*

Stemming

- Reduce terms to their “roots” before indexing
- “Stemming” suggest crude affix chopping
  - language dependent
  - e.g., ***automate(s), automatic, automation*** all reduced to ***automat***.



for example compressed and compression are both accepted as equivalent to compress.

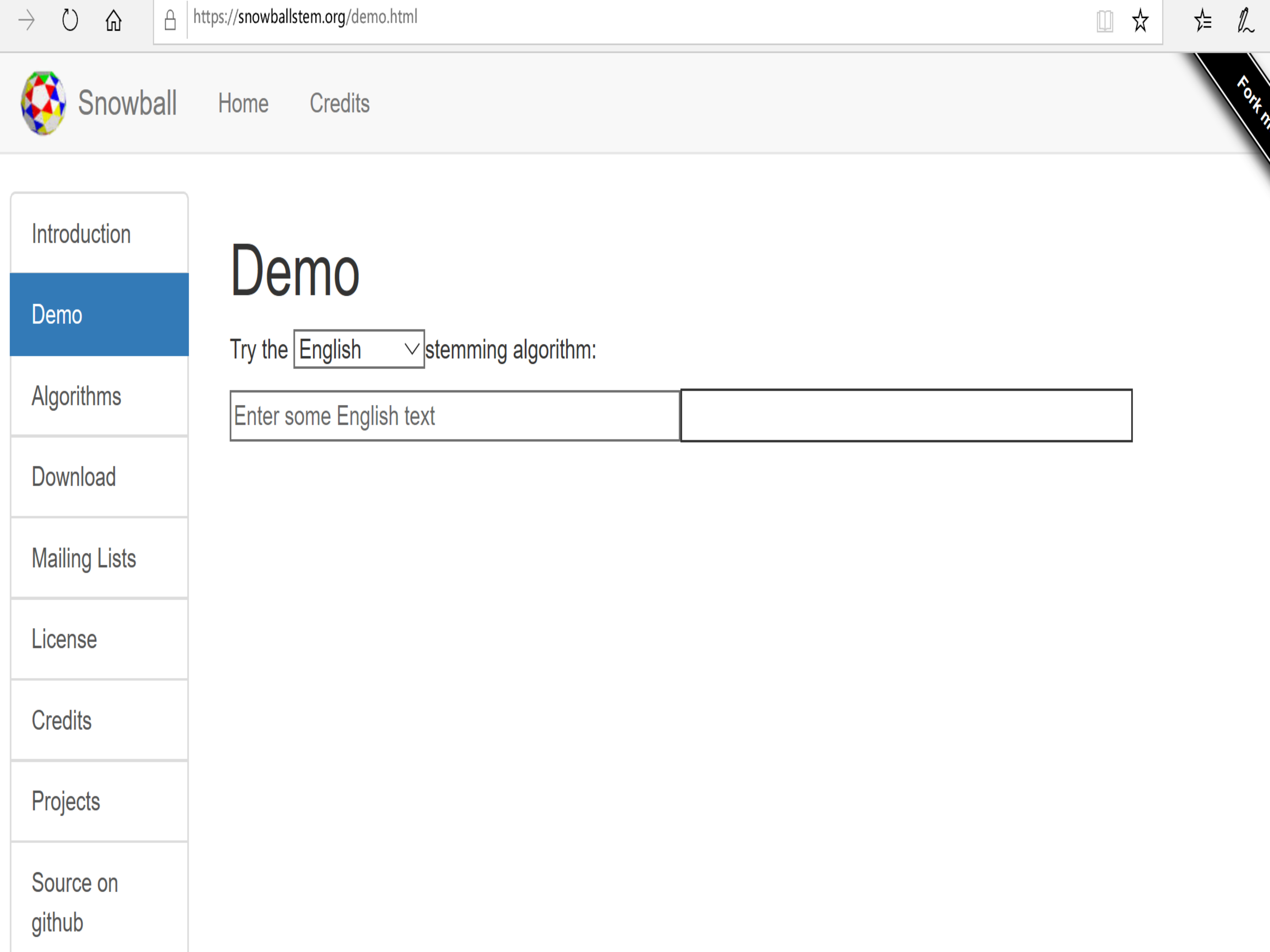
for exampl compress and compress ar both accept as equival to compress

Porter’s algorithm

- Commonest algorithm for stemming English

Porter’s stemmer:
参考 http://www.tartarus.org/~martin/PorterStemmer/
参考 https://snowballstem.org/
DEMO: https://snowballstem.org/demo.html

- Do stemming and other normalizations help?
  - English: very mixed results. Helps recall for some queries but harms precision on others
    - E.g., operative (dentistry) ⇒ oper
  - Definitely useful for Spanish, German, Finnish, …
    - 30% performance gains for Finnish!

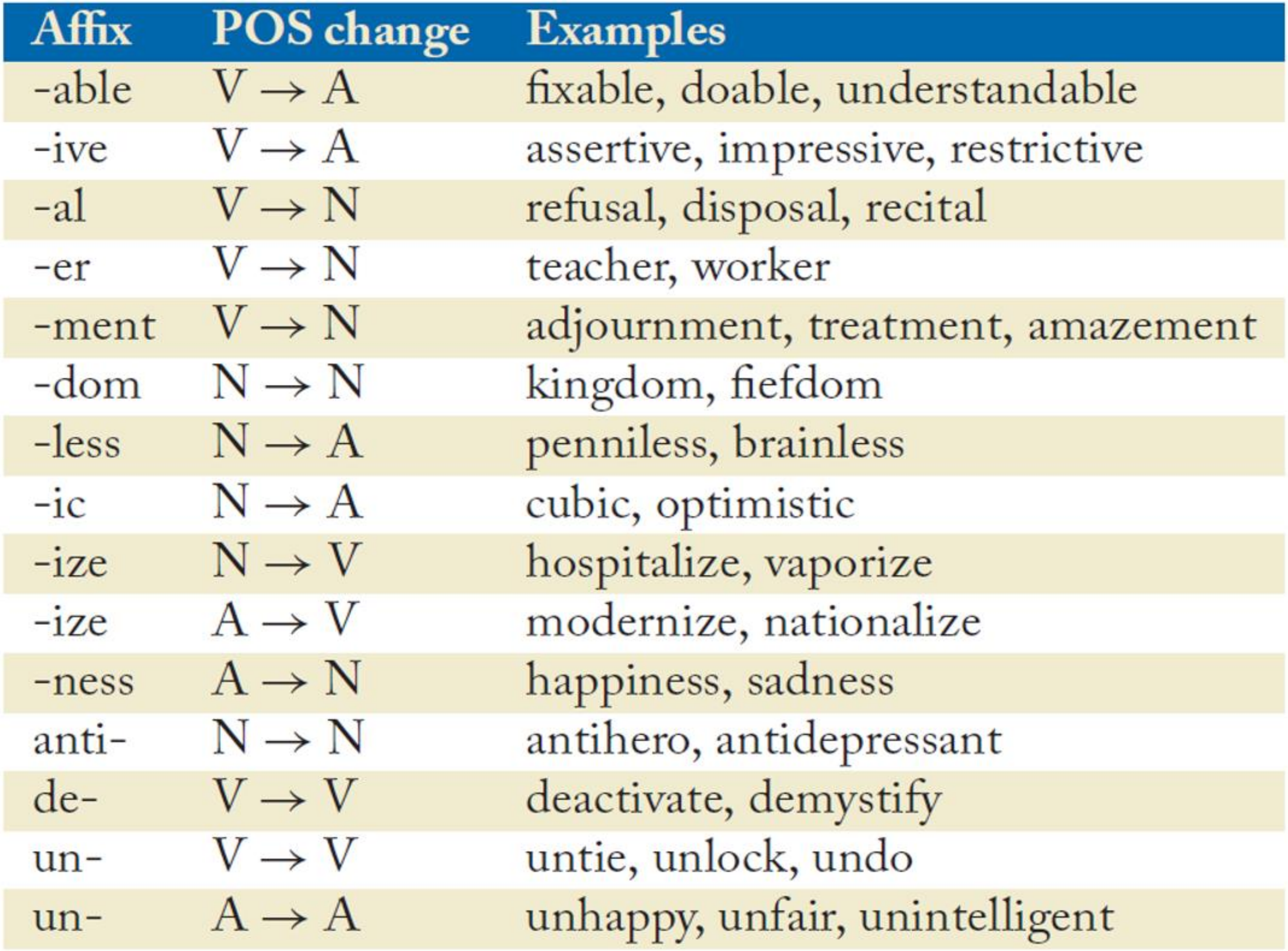


Demo

Try the English stemming algorithm:

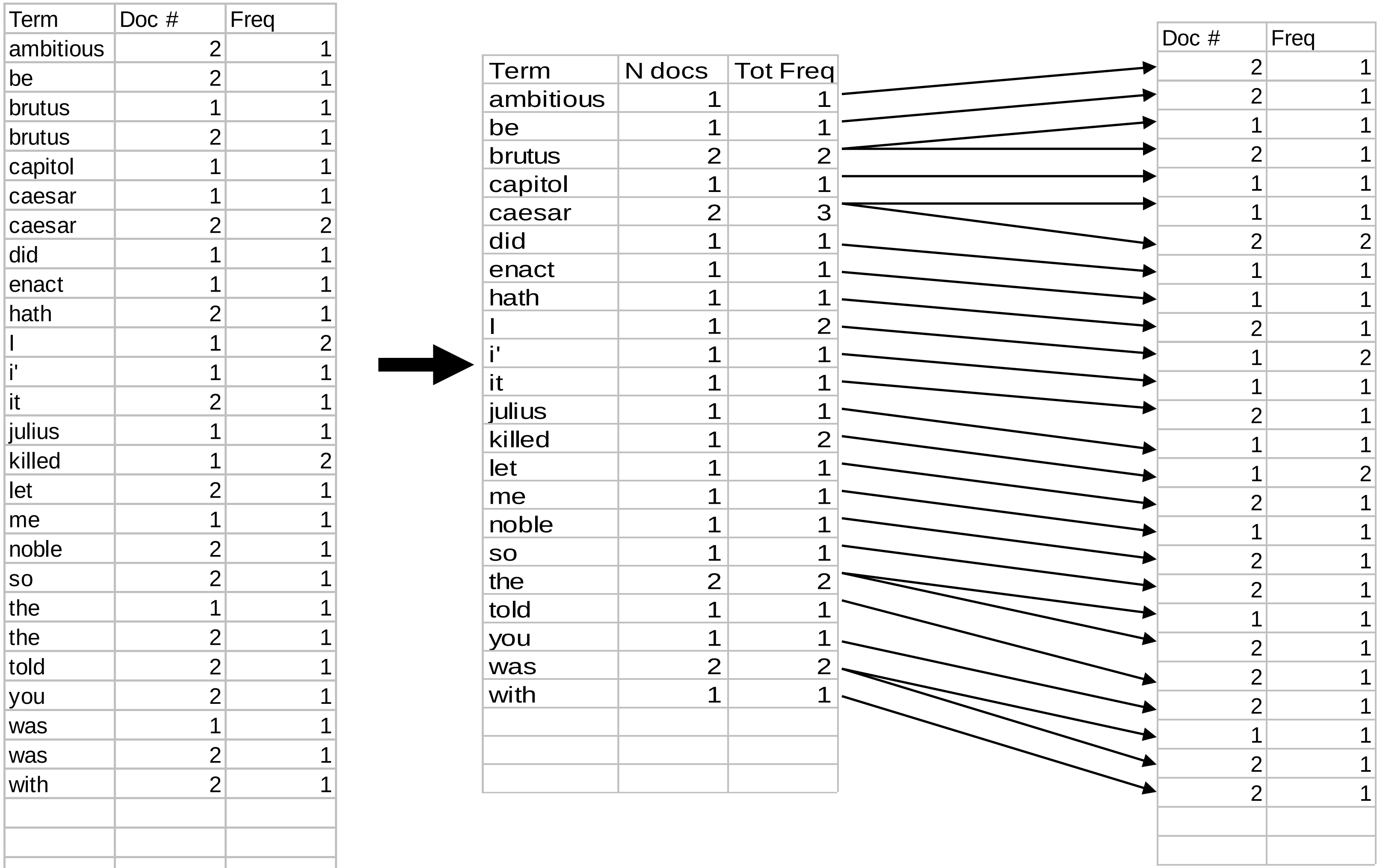
Enter some English text

A sample of English derivational affixes [O'Grady *et al.*, 2010, 124]



Affix	POS change	Examples
-able	V → A	fixable, doable, understandable
-ive	V → A	assertive, impressive, restrictive
-al	V → N	refusal, disposal, recital
-er	V → N	teacher, worker
-ment	V → N	adjournment, treatment, amazement
-dom	N → N	kingdom, fiefdom
-less	N → A	penniless, brainless
-ic	N → A	cubic, optimistic
-ize	N → V	hospitalize, vaporize
-ize	A → V	modernize, nationalize
-ness	A → N	happiness, sadness
anti-	N → N	antihero, antidepressant
de-	V → V	deactivate, demystify
un-	V → V	untie, unlock, undo
un-	A → A	unhappy, unfair, unintelligent

2.4 对倒排文件的进一步考察



Term	Doc #	Freq
ambitious	2	1
be	2	1
brutus	1	1
brutus	2	1
capitol	1	1
caesar	1	1
caesar	2	2
did	1	1
enact	1	1
hath	2	1
I	1	2
i'	1	1
it	2	1
julius	1	1
killed	1	2
let	2	1
me	1	1
noble	2	1
so	2	1
the	1	1
the	2	1
told	2	1
you	2	1
was	1	1
was	2	1
with	2	1

Term	N docs	Tot Freq
ambitious	1	1
be	1	1
brutus	2	2
capitol	1	1
caesar	2	3
did	1	1
enact	1	1
hath	1	1
I	1	2
i'	1	1
it	1	1
julius	1	1
killed	1	2
let	1	1
me	1	1
noble	1	1
so	1	1
the	2	2
told	1	1
you	1	1
was	2	2
with	1	1

Doc #	Freq
2	1
2	1
1	1
2	1
1	1
1	1
2	2
1	1
1	1
2	1
1	2
1	1
2	1
1	1
1	2
2	1
1	1
2	1
2	1
1	1
2	1
2	1
2	1
1	1
2	1
2	1

The file is commonly split into a *Dictionary* and a *Postings List*

2.4 对倒排文件的进一步考察

For the Dictionary

- How big is the term vocabulary?

  That is, how many distinct words are there?

- In practice, the vocabulary will keep growing with the collection size

Vocabulary vs. collection size

- Heaps' law: $M = kT^b$
- M is the size of the vocabulary, T is the number of tokens in the collection
- Typical values: $30 \leq k \leq 100$ and $b \approx 0.5$
- In a log-log plot of vocabulary size M vs. T, Heaps' law predicts a line with slope about ½
  - It is the simplest possible relationship between the two in log-log space
  - An empirical finding ("empirical law")

Heaps' Law

For RCV1, the dashed line

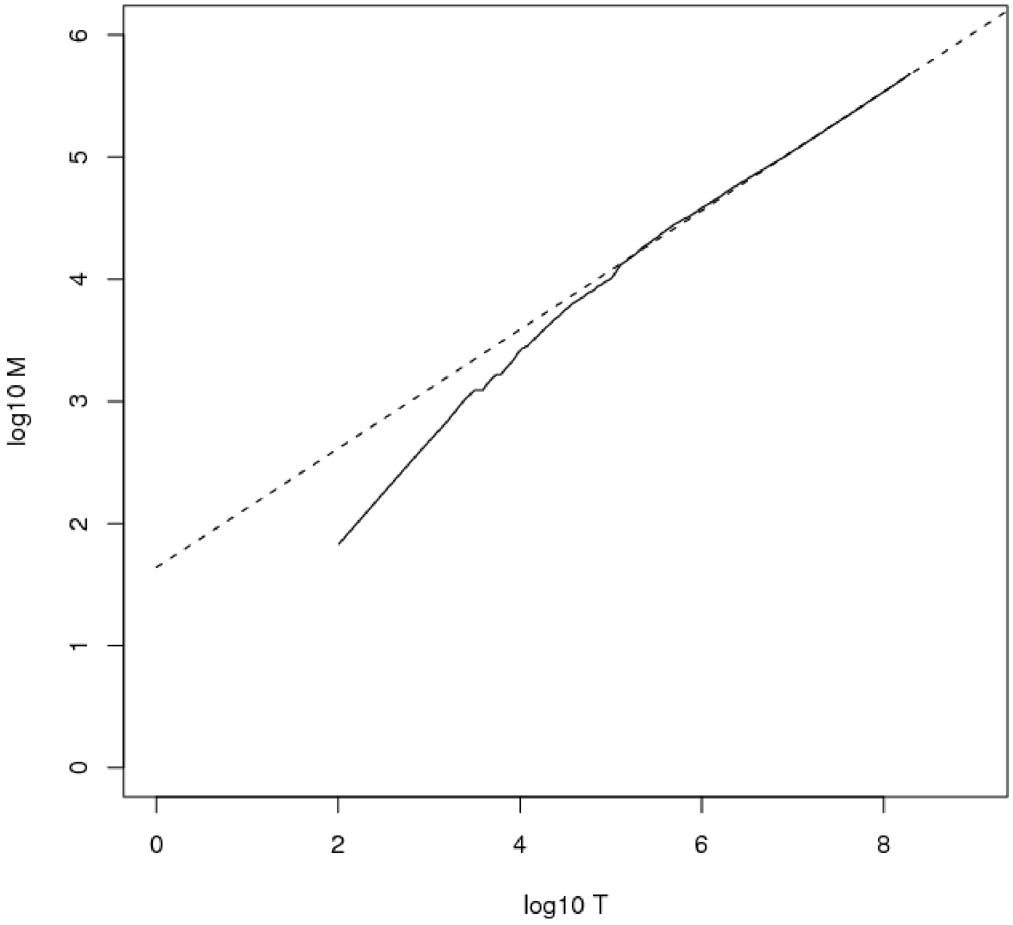
$\log_{10} M = 0.49 \log_{10} T + 1.64$

is the best least squares fit.

Thus, $M = 10^{1.64} T^{0.49}$ so $k = 10^{1.64} \approx 44$ and $b = 0.49$.

Good empirical fit for Reuters RCV1 !

For first 1,000,020 tokens, law predicts 38,323 terms; actually, 38,365 terms



作业

- 分别寻找任意一个针对英文的Stemmer和Lemmatizer(也可以用课堂PPT上推荐的)，任意选择风格不一致的三个文章小片段，分别做stemming和lemmatization， 观察结果并做比较，进一步地，对其对信息检索可能造成的影响进行分析。