https://vvtesh.sarahah.com/

Information Retrieval

Venkatesh Vinayakarao

Term: Aug – Sep, 2019
Chennai Mathematical Institute

அட பாடல் போல தேடல் கூட ஒரு சுகமே
Ada Padal Pola Thedal Kooda Oru Sugame
Search, like a song, is also a joy.

- From the movie, Thulladha Manamum Thullum. Lyrics by Vaali.

Review

Documents

Terms	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

"Brutus and Caesar and not Calpurnia"

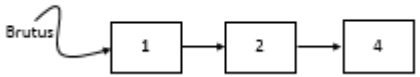
1	1	0	1	0	0
1	1	0	1	1	1
1	0	1	1	1	1
AND					
1	0	0	1	0	0

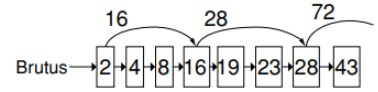
Document 1 and 4 satisfy our query.

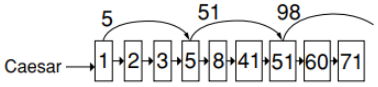
One (bad) Approach

- First match the **term** IIIT.
  - Filter out documents that contain this term.
- Next match the **term** Sri.
  - Filter out documents that contain this term.
- Next match the **term** City.
  - Filter out documents that contain this term.

~~int[] A = {1,1,1};~~





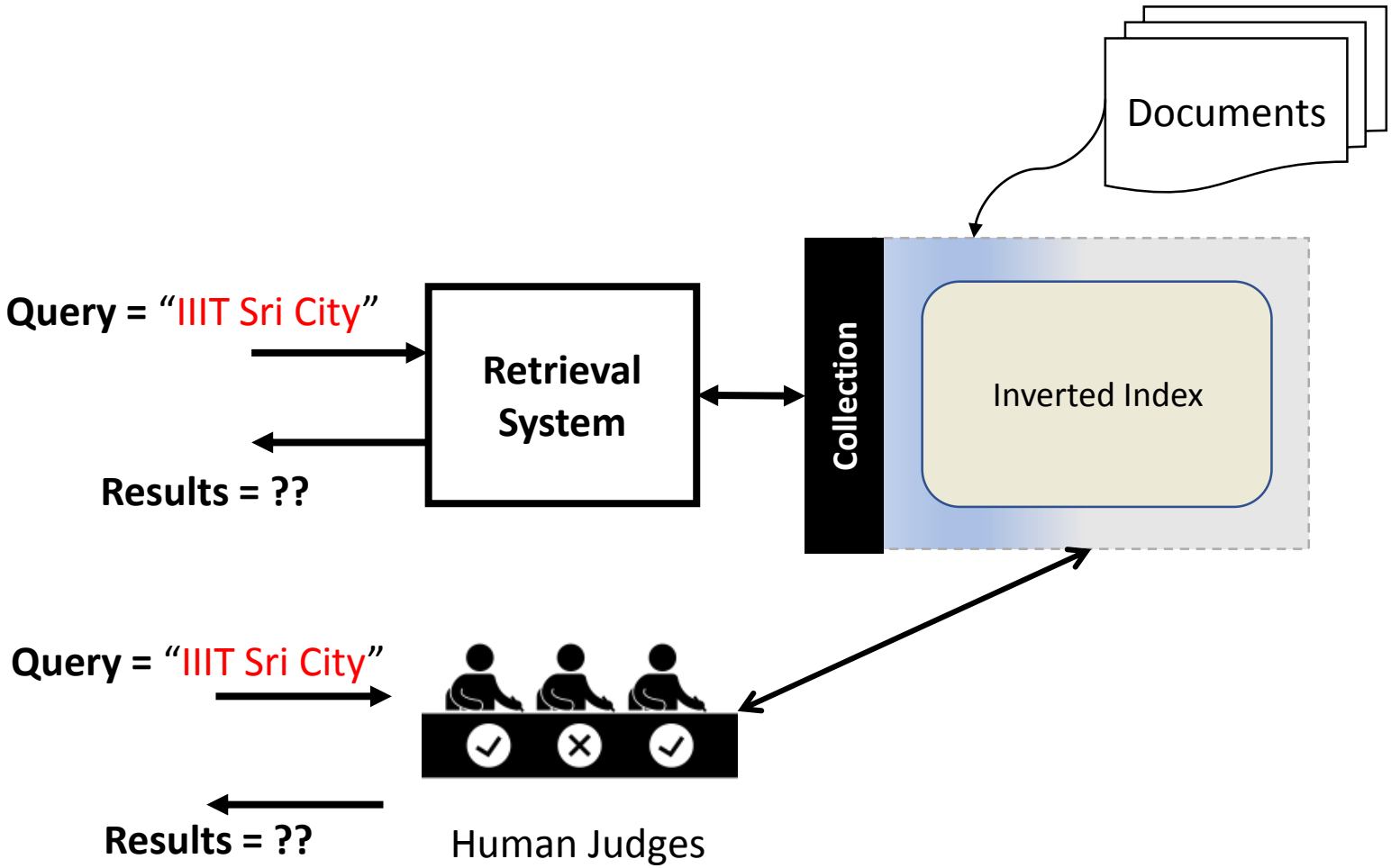


Evaluation

How Good is Our System?

- A **collection** having the following contents
  - d1: IIIT ALLAHABAD
  - d2: IIIT DELHI
  - d3: IIIT GUWAHATI
  - d4: ISI
  - d5: IIIT SRI CITY
  - d6: KREA SRI CITY
- **Query** is
  - SRI CITY
- **Result** is
  - IIIT SRI CITY
  - KREA SRI CITY

Evaluation


Documents
Query = "IIIT Sri City"
Retrieval System
Results = ??
Collection
Inverted Index
Query = "IIIT Sri City"
Results = ??
Human Judges

How Good is Our System?

- A **collection** having the following contents
  - d1: IIIT ALLAHABAD
  - d2: IIIT DELHI
  - d3: IIIT GUWAHATI
  - d4: ISI
  - d5: IIIT SRI CITY
  - d6: KREA SRI CITY
- **Query** is
  - IIIT
- **Result** is
  - IIIT SRI CITY
  - KREA SRI CITY

Objective

We want all relevant documents and only relevant documents

Relevance

- How many **relevant** documents?
  - Four (IIIT SRI CITY, IIIT ALLAHABAD, IIIT DELHI, IIIT GUWAHATI)
- How many **retrieved** documents?
  - Two (IIIT SRI CITY, KREA SRI CITY)

How to quantify the “goodness” of our system?

Terminology

- Documents we see in results are "**positive**"
  - Positive
    - + IIIT SRI CITY,
    - + KREA SRI CITY
  - Negative
    - - IIIT ALLAHABAD
    - - IIIT DELHI
    - - IIIT GUWAHATI
    - - ISI

Terminology

- Documents that we correctly classify are "**true**"
  - Positive
    - + IIIT SRI CITY (true)
    - + KREA SRI CITY
  - Negative
    - - IIIT ALLAHABAD
    - - IIIT DELHI
    - - IIIT GUWAHATI
    - - ISI (true)

Here, query is "IIIT"

Quiz

- All retrieved results =
  1. tp + fp
  2. tp + fn
  3. tn + fp
  4. tn + fn

Legend

tp = true positive

tn = true negative

fp = false positive

fn = false negative

Quiz

- All retrieved results =
  1. tp + fp
  2. tp + fn
  3. tn + fp
  4. tn + fn

Legend

tp = true positive

tn = true negative

fp = false positive

fn = false negative

Quiz

- All relevant results =
  1. tp + fp
  2. tp + fn
  3. tn + fp
  4. tn + fn

Legend

tp = true positive

tn = true negative

fp = false positive

fn = false negative

Quiz

- All relevant results =
  1. tp + fp
  2. tp + fn
  3. tn + fp
  4. tn + fn

Legend

tp = true positive

tn = true negative

fp = false positive

fn = false negative

You have 100% Precision

- Everything you retrieved were relevant.
  - tp + fp = tp
  - fp = 0

You have 100% Recall when

- You retrieved everything that were relevant. (Note: You could have retrieved more).
  - fn = 0
  - tp = all relevant documents

Quiz

- R refers to Relevant Document
- N refers to Nonrelevant Document.
- Collection has 10,000 documents.
- Assume that there are 8 relevant documents in total in the collection. Calculate Precision and Recall.
- Retrieved Documents:

  RRNNN NNN**R**N **R**NNN**R** NNNN**R**

Precision and Recall

- Precision = 6/20
- Recall = 6/8

Precision and Recall

Precision: fraction of retrieved docs that are relevant = P(relevant|retrieved)

Recall: fraction of relevant docs that are retrieved

= P(retrieved|relevant)

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Precision P = tp/(tp + fp)
- Recall R = tp/(tp + fn)

Exercise

Suppose, a document is relevant only if both judges agree that it is relevant. Assume (0 = nonrelevant, 1 = relevant). What is the Precision and Recall?

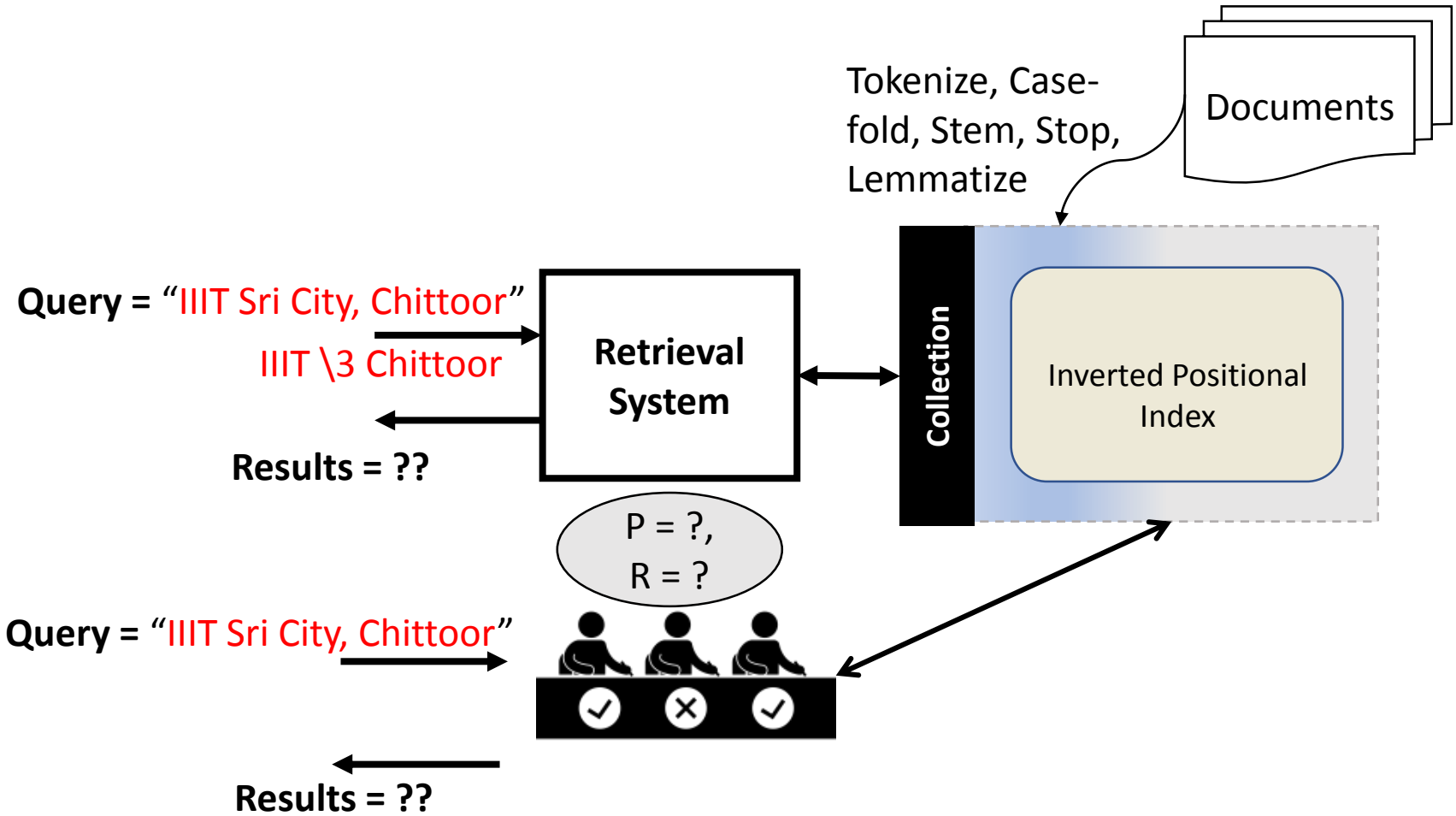
Query = "Taj"

Document ID	Judge 1	Judge 2	Our System
d1 = Bru	0	0	Retrieved
d2 = 3Roses	0	0	No
d3 = Taj	1	1	Retrieved
d4 = Taj Tea	1	1	No
d5 = Taj Mahal	1	0	No

Answer

- Precision = 1/2
- Recall = 1/2

How does a Search Engine Work?



Westlaw.com

- A commercial boolean legal search service [1975].

A Westlaw Query

disab! /p access! /s work-site work-place (employment /3 place)

Information Need = Requirements for disabled people to be able to access a workplace.

- Conventions
  - work-site matches worksite, work-site, or work site.
  - disab! matches all words starting with disab!
  - space is disjunction.
  - \p => match within paragraph, \s => match within sentence, \3 => match within 3 words

Quiz

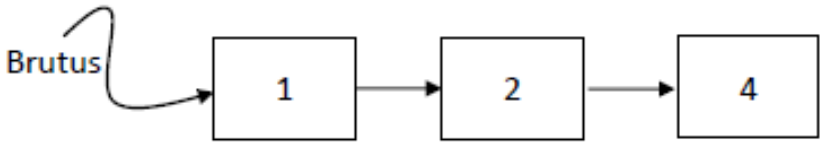
- Choose the best answer: AND operators in boolean search tends to produce
  - high precision and low recall
  - low precision and high recall

Query Processing Order

Term Document Matrix & Postings List

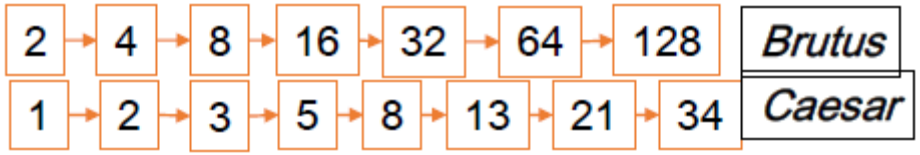
Documents

Terms	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0



Query Processing

- Brutus AND Caeser



- Which document(s) should result?
- How many comparisons did you do?

Query Processing

- X AND Y

X	1	2	3	4	5	6	7	8
Y	1	2	3					

- Which document(s) should result?
- How many comparisons did you do?
- Is there any way we could get:
  - |result| > min(|x|, |y|)
    - No!

Query Processing

- X AND Y

X								
Y								

- If |X| = 3, |Y| = 5, Can you fill the boxes in two different ways such that the number of comparisons are different?

Query Processing

- X AND Y

X	1	3	5					
Y	1	2	3	4	5			

- No. of Comparisons = |(1,1), (3,2), (3,3), (5,4), (5,5)| = 5.

Query Processing

- X AND Y

X	1	2	3					
Y	1	2	3	4	5			

- No. of Comparisons = |(1,1), (2,2), (3,3)| = 3.

Query Processing

- X AND Y

X								
Y								

- If |X| = 3, |Y| = 5, Can you fill the boxes in two different ways such that the number of comparisons are ~~different~~ maximum?

Query Processing

- X AND Y

X	1	2	8					
Y	4	5	6	7	8			

- No. of Comparisons = |(1,4),(2,4),(8,4),(8,5),(8,6),(8,7),(8,8)| = 7

- Can you do any better?

Query Processing

- X AND Y
- Min. No. of Comparisons =3

X	1	2	3					
Y	1	2	3	4	5			

- Max. No. of Comparisons = 7

X	1	2	8					
Y	4	5	6	7	8			

- Is there a better answer?

Query Processing

- Query: Brutus AND Caeser AND Calpurnia
- Assumption:
  - Brutus appears in 10 documents.
  - Caeser appears in 5 documents.
  - Calpurnia appears in 3 documents.
- How many comparisons?
  - Option 1: Merge Brutus AND Caeser first. Merge the result with Calpurnia.
  - Option 2: Merge Caeser AND Calpurnia first. Merge the result with Brutus.

Query Processing Order

- Option 1: Merge Brutus AND Caeser first. Merge the result with Calpurnia.
  - Brutus AND Caeser: In worst case, requires (10 + 5) = 15 comparisons.
  - Result AND Calpurnia: In worst case, requires (5 + 3) = 8 comparisons.
  - Therefore, requires 15 + 8 = 23 comparisons.
- Option 2: Merge Caeser AND Calpurnia first. Merge the result with Brutus.
  - Caeser AND Calpurnia: In worst case, requires 5 + 3 = 8 comparisons.
  - Result AND Brutus: In worst case, requires 3 + 10 = 13 comparisons.
  - Therefore, 13 + 8 = 21 comparisons.

*We approximate worst case comparisons to x+y for convenience.

Process in increasing order by frequency.

Do you now see why we store <u>frequency</u> with our Dictionary terms?

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2

Quiz

What is the best order of processing "eyes and skies and trees"?

Term	Postings size
eyes	213312
kaleidoscope	87009
marmalade	107913
skies	271658
tangerine	46653
trees	316812

What about "(eyes or skies) and (trees or tangerine) and (marmalade or kaleidoscope)"?

Find the query processing order for
(A OR B) AND (C OR D) AND (E OR F)

(A OR B) AND (C OR D) AND (E OR F)

- Let x = Freq(A) + Freq(B)
- Let y = Freq(C) + Freq(D)
- Let z = Freq(E) + Freq(F)

- A OR B leaves us with A union B items. In worst case, we have freq(A) + freq(B) items.

- We know how to solve (x AND y AND z)

Quiz

Term	Postings size
eyes	213312
kaleidoscope	87009
marmalade	107913
skies	271658
tangerine	46653
trees	316812

What about (tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)**?**

Answer: ((kaleidoscope OR eyes) AND (tangerine OR trees)) AND (marmalade or skies)

Indexing

How to Index?

Captain Haddock

What is a Document?

- Some systems store a single email in multiple files. Is each file a document?
- Some files can contain multiple documents (as in XML, Zip).

Tokens Vs. Terms

- Tokens
  - Input: Friends, Romans, Countrymen, lend me your ears.
  - Output:

Friends	Romans	Countrymen	lend	me	your	ears

  - Sequence of characters → Semantic Units
    - Throw away “less important” parts (like punctuation)
- Terms
  - Indexed by the IR system

Quiz

- Tokenize O’Neil Can’t study.

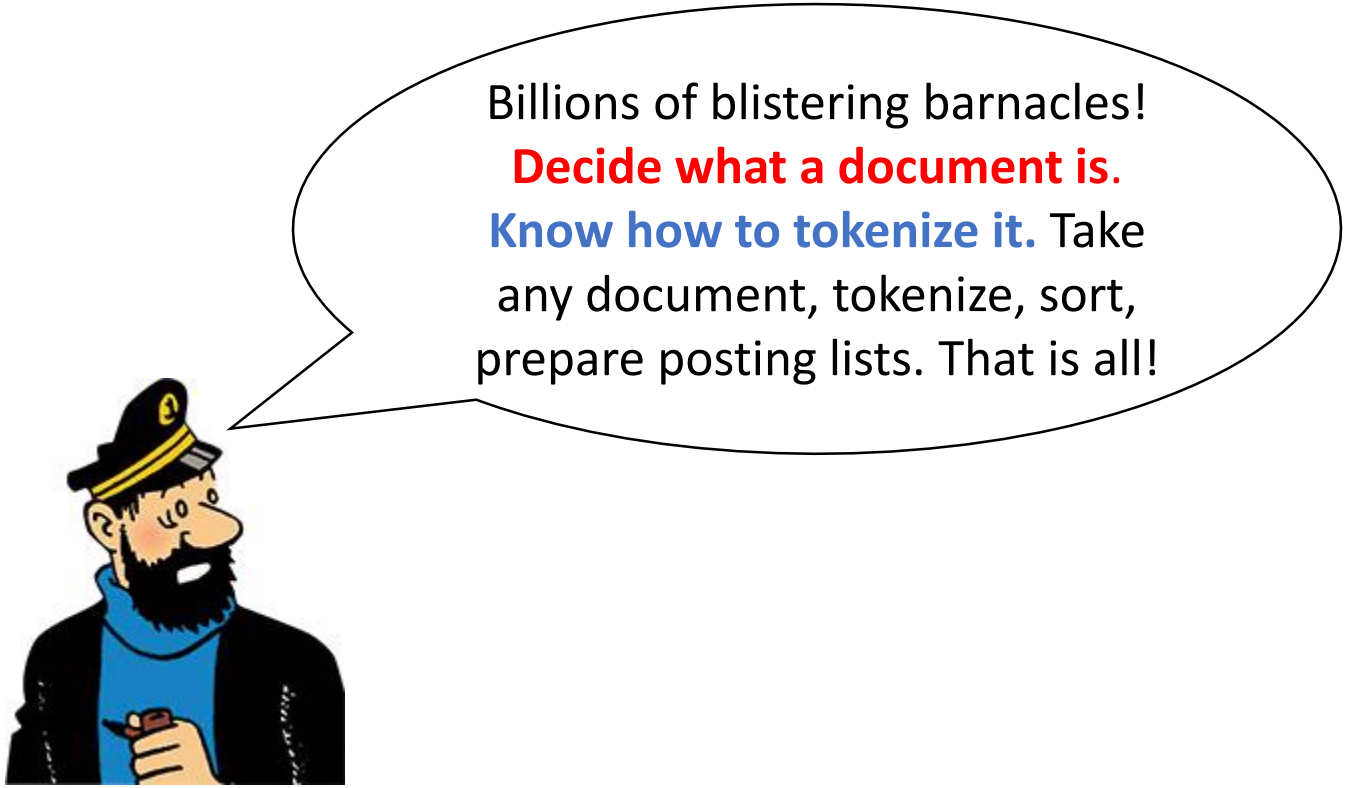
O’Neil	Can’t	study

What if we tokenize based on ‘ ?

O	Neil	Can	t	study

O	Neil	Can’t	study

How to Index?


Billions of blistering barnacles!
Decide what a document is.
Know how to tokenize it. Take any document, tokenize, sort, prepare posting lists. That is all!

Which Tokens to Index?

- Which tokens are interesting?

It is difficult to imagine living without search engines

Stop Word Removal

difficult imagine living search engines

- it, is, to, without are “Stop Words” for us here.

How to Index?


Billions of blue blistering barnacles! Decide what a document is. Know how to tokenize it. Prepare a stop words list. Take any document, tokenize, remove stop words, sort, prepare posting lists. That is all!

Token Normalization

- Equivalence Classes
  - (case folding) window, windows, Windows, Window → window
  - anti-theft, antitheft, anti theft → antitheft
  - color, colour → color

Billions of bilious blue blistering barnacles! **Decide what a document is. Know how to tokenize it. Prepare a stop words list.** Take any document, tokenize, **normalize**, **remove stop words**, sort, prepare posting lists. That is all!

Normalization Challenges

- We lose the meaning if we normalize incorrectly:
  - C.A.T is not cat
  - Bush may be a person name. Need to be careful with proper nouns.
- Is TrueCasing a potential solution?
  - TrueCasing
    - Convert words at beginning of a sentence to lowercase.
    - Leave the rest capitalized.
- Usually, we lowercase everything.

Stemming and Lemmatization

- Stemming (chop the ends)
  - going → go, analysis → analys (Need not result in a dictionary word)
- Lemmatization
  - Return the dictionary form of the root word (lemma)
    - saw → see.
- More Examples
  - am, are, is → be
  - car, cars, car's, cars' → car
  - democrat, democratic, democracy, democratization → democrat

Porter Stemmer

- Multiple phases of rule-based refinement

Rule	Example
SSES → SS	caresses → caress
IES → I	ponies → poni
SS → SS	caress → caress
S →	cats → cat
(m > 1) EMENT →	replacement → replac (does not apply to **cement**)

word measure

Stemming Examples

Stemmer	Text
Porter	Such an analysis can reveal features that are not easil visible from the variations in the individual genes and can lead to a picture of expression that is more
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
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

Issues in Stemming

- Stemmers are not perfect!
- Overstemming
  - Too many characters are cut off from the word
  - Example: university, universal → univers
- Understemming
  - Example: data → dat, datum → datu. Ideally, we would like the result to be the same for both.

How to Index?

Billions of bilious blue blistering barnacles! **Decide what a document is. Know how to tokenize it. Prepare a stop words list.** Take any document, tokenize, **normalize**, **remove stop words**, **stem/lemmatize**, sort, prepare posting lists. That is all!

Quiz

- Can you tokenize the following?
  - **반갑습니다**
    - (Korean for "*Nice to meet you*")
  - **Bundesausbildungsförderungsgesetz**
    - A German compound word for "*Federal Education and Training Act*")
- Can you think of a case where splitting with white space is bad?
  - Los Angeles, New Delhi, IT Park