



CSEN1076: NATURAL LANGUAGE PROCESSING AND INFORMATION RETRIEVAL

LECTURE 1 – INTRODUCTION & INFORMATION RETRIEVAL I

MERVAT ABUELKHEIR

FIRST, USEFUL INFORMATION

Course Name Natural Language Processing and Information Retrieval

My Email mervat.abuelkheir@guc.edu.eg

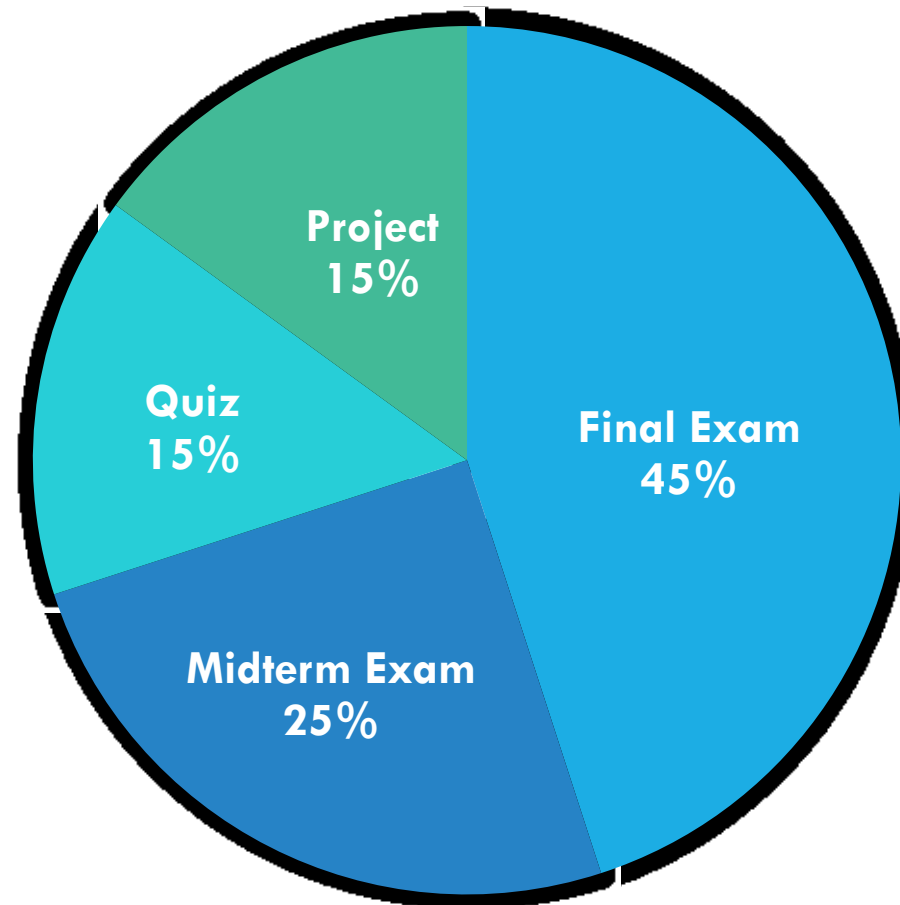
TA Mohamed Abdelfattah

Textbooks **Introduction to Information Retrieval**, Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. <https://nlp.stanford.edu/IR-book/>

Natural Language Processing with Python, Steven Bird, Ewan Klein, and Edward Loper. <http://www.nltk.org/book/>

Optional **Speech and Language Processing**, Daniel Jurafsky and James Martin, Prentice Hall, 2000.

COURSE GRADING



REQUIRED TOOLS

All of the following tools have to be installed on your laptops before your next tutorial:

- Python 3.X
- Jupyter (Python IDE)
- NLTK (Natural Language Processing Toolkit)
- Seaborn (Statistical Data Visualization)
- Sublime (Text Editor)

Don't forget to bring your laptop with you on every lab session

It would be great to learn about Python before your next tutorial

IR? NLP?

IR — Extract

Finding resources (of unstructured nature) that are relevant to an information need from a large collection of resources



"If you trust your search engine more than you trust me, maybe you should switch doctors."

NLP — Interact

Creating systems that efficiently process texts and make their information accessible to computer applications



THE KEY IS IN “NATURAL”

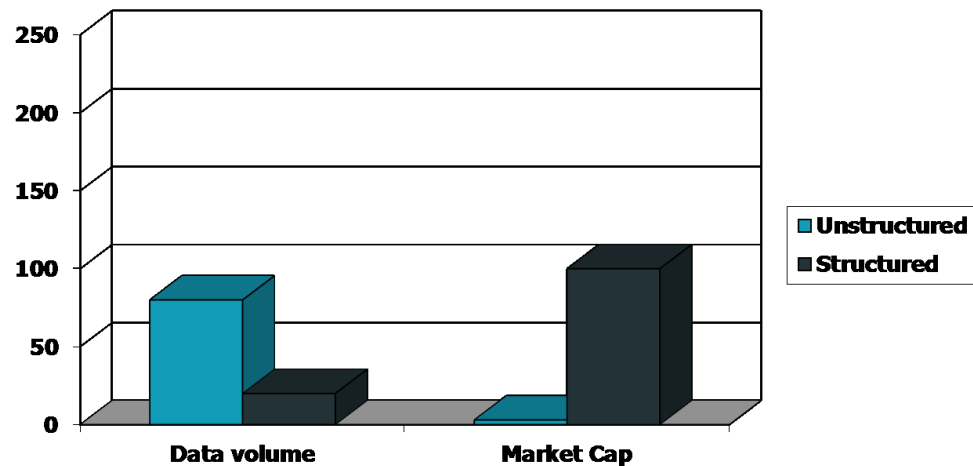
IR and NLP are not concerned with structured data

There is SQL for that!

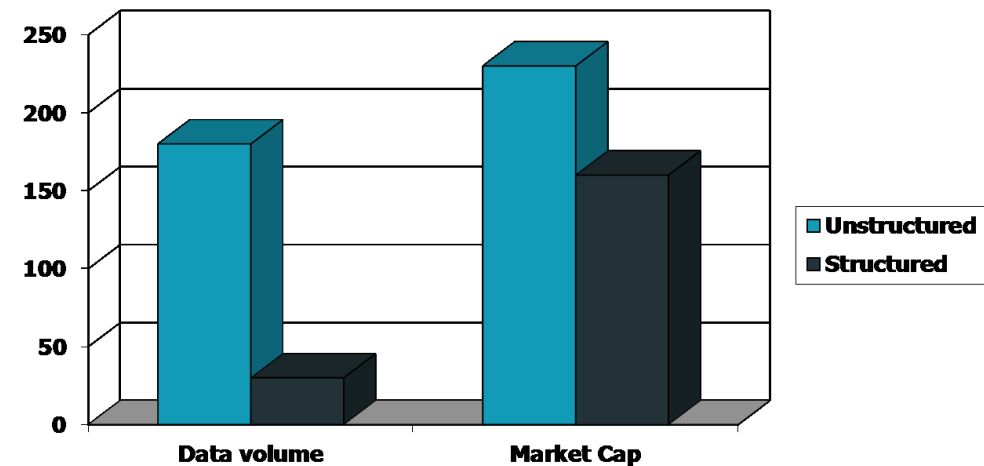
We are not looking for “**exact**” results, but for – hopefully – “**relevant**” results

THINGS HAVE CHANGED

Unstructured (text) vs. structured (database) data in the mid-nineties



Unstructured (text) vs. structured (database) data today



IR EXAMPLE APPLICATIONS

Web Search (duh!)

The screenshot shows a Google search for "nlp". The search bar at the top contains "nlp" and the Google logo. Below the search bar, there are tabs for "All", "Images", "Videos", "News", "Books", and "More". The "All" tab is selected. Below the tabs, it says "About 39,700,000 results (0.64 seconds)".

The search results include:

- NLP Practitioner Training | Online Courses**
An advertisement for www.robertsimiccoachinginstitute.com/ with a phone number +971 50 685 4971. The text describes their approach to change as simple, practical, and effective, offering personal coaching and live demonstrations.
- Life Coaching Courses**
Text: "Don't just learn of Life Coaching, Become an Excellent Life Coach!"
- Contact Us**
Text: "Get In Touch with RSCI & Experience The Difference For Yourself!"
- Neuro-linguistic programming - Wikipedia**
Link: https://en.wikipedia.org/wiki/Neuro-linguistic_programming
Text: "Neuro-linguistic programming (NLP) is an approach to communication, personal development, and psychotherapy created by Richard Bandler and John Grinder ... Methods of neuro-linguistic ... · Richard Bandler · John Grinder"
- What is NLP? - NLP.com**
Link: www.nlp.com/what-is-nlp/
Text: "NLP stands for Neuro-Linguistic Programming. Neuro refers to your neurology; Linguistic refers to language; programming refers to how that neural language functions. ... In NLP, we have a saying: the conscious mind is the goal setter, and the unconscious mind is the goal getter."

On the right side of the search results, there is a "More images" section showing various diagrams and illustrations related to NLP, including a brain diagram, a tree diagram, and a person diagram.

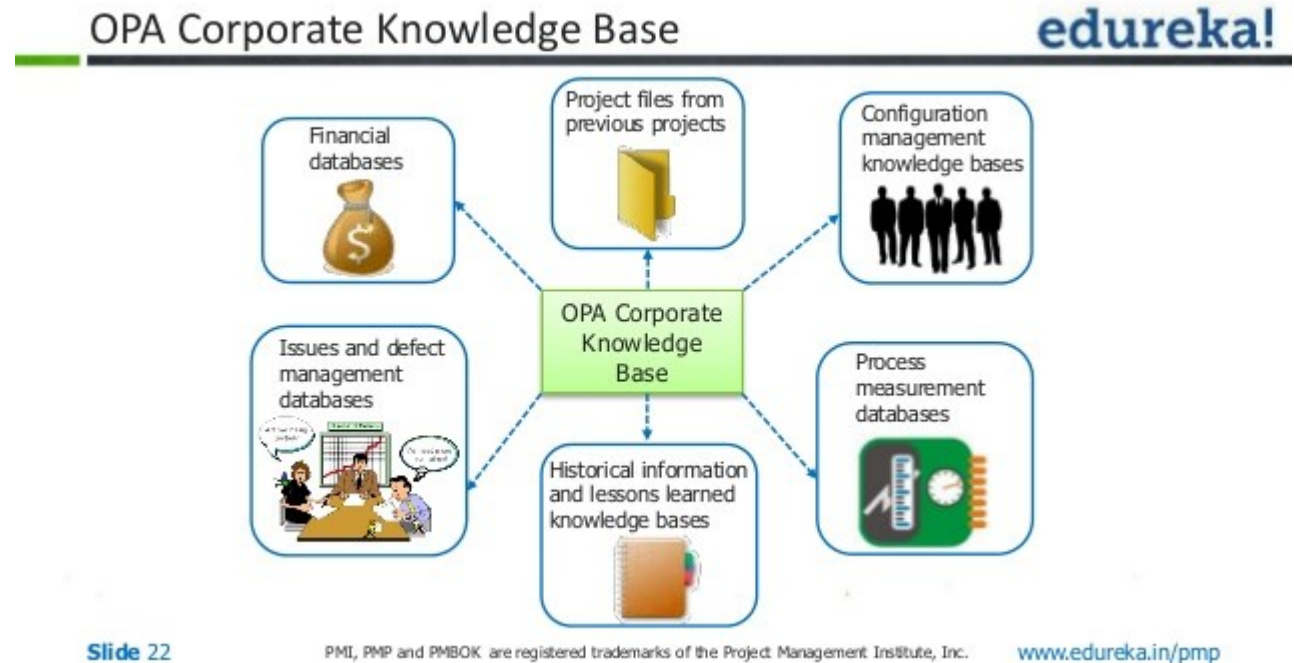
Below the search results, there is a "People also ask" section with the following questions:

- What is the NLP technique?
- What is NLP therapy used for?
- What NLP stands for?
- How does NLP really work?

At the bottom right of the "People also ask" section, there is a "Feedback" link.

IR EXAMPLE APPLICATIONS

Corporate Knowledgebase



IR EXAMPLE APPLICATIONS

Legal Information Retrieval

The screenshot shows the WestlawNext search results page for the query "employer facebook password". The interface includes a search bar at the top with the query and a "SEARCH" button. On the left, there is a sidebar with various filters and a table of results. The main content area displays the search results, including a list of cases and a detailed view of the selected case, "2. U.S. v. Nosal".

WestlawNext powered by WestSearch™

SPAR LISA | Folders | History | Alerts | Sign Off

Q - employer facebook password All State & Federal SEARCH advanced false claims act (10)

Jury Verdicts & Settlements 5,875
Proposed & Enacted Legislation 2,591
Proposed & Adopted Regulations 781
Arbitration Materials 5,293
Public Records 2,391
All Results 30,689

NARROW:
Apply Filters Cancel

Search within results
Q -

Jurisdiction
* Federal 22
* State 7
All

Date
All

Reported Status
Reported 21
Unreported 8

Topic
Criminal 21
Civil 19
Employment & Labor 12
Intellectual Property 11
Antitrust 6

2. U.S. v. Nosal
United States District Court, N.D. California. March 12, 2013. 930 F.Supp.2d 1051. 2013 WL 978226. CR-08-0237 EMC

CRIMINAL JUSTICE - Computer Crimes. Indictment sufficiently alleged unauthorized access of a protected computer under the Computer Fraud and Abuse Act.

...For defendant to have acted without authorization, within the meaning of the Computer Fraud and Abuse Act (CFAA), in accessing confidential computer database of his former employer by using the computer username and password of his former assistant who was still working for the employer, defendant was not required to have obtained his assistant's password illegally or without her consent, the assistant allegedly willingly provided her access credentials to defendant. 18 U.S.C.A. § 1030(a)(4, 6)...

... Furthermore, allowing such person to use one's password permits them to access the user's Facebook account containing the user's personal account and information; it does not allow access to any Facebook trade secrets...

... Use of another's password "avoids" and "bypasses" the technological measure of password protection...

... Holdings: The District Court, Edward M. Chen, J., held that: (1) government was not required to prove that defendant circumvented technological access barriers; (2) even if it was, indictment sufficiently alleged defendant circumvented such barriers; (3) to support charge that defendant used his former assistant's password to access employer's computer system, defendant was not required to have obtained his assistant's password illegally or without her consent; (4) allegation assistant accessed the system and then handed computer terminal to defendants associate was sufficient to allege unauthorized access...

3. Ehling v. Monmouth-Ocean Hosp. Service Corp.
United States District Court, D. New Jersey. August 20, 2013. 961 F.Supp.2d 659. 2013 WL 4436539. 2:11-CV-03305 WJM

ENERGY AND UTILITIES - Telecommunications. Wall posts on social networking website fell within purview of Stored Communications Act (SCA).

... Caruso never had the password to Ronco's Facebook account, Plaintiffs Facebook account, or any other employee's Facebook account...

... Caruso never had the password to Ronco's Facebook account, Plaintiffs Facebook account, or any other employee's Facebook account...

... To create Facebook wall posts, Facebook users transmit writing, images, or other data via the Internet from their computers or mobile devices to Facebook's servers...

... It is undisputed that Ronco was a Facebook user. Plaintiff acknowledged that she added Ronco as a Facebook friend and posted on Ronco's Facebook wall...

Facebook page is in effect being asked to verify the existence of his Facebook account, his control over it, and its authenticity, all admissions that carry Fifth Amendment protection...

BE WARY OF DEMANDING ACCESS TO EMPLOYEES' SOCIAL MEDIA ACCOUNTS

20 No. 4 Ariz. Emp. L. Letter 1 September 2013
Arizona Employment Law Letter
...An Arizona employer requires Jane, an employee, to disclose her Facebook username and password...

YOUR PASSWORD OR YOUR PAYCHECK?: A JOB APPLICANT'S MURKY RIGHT TO SOCIAL MEDIA PRIVACY

16 No. 3 J. Internet L. 1 September, 2012
Journal of Internet Law
... Emma Barnett, "Facebook Passwords: Fair Game in Job Interviews," D. Telegraph, Mar. 23, 2012, <http://www.telegraph.co.uk/technology/facebook/916235/facebook-passwords-fair-game-in-job-interviews.html> (Reporting one instance of a retail employer in the UK asking an employee for a password); "U.S. Joins Debate On Employers Demanding Passwords," CBC, Apr. 24, 2012, <http://www.huffingtonpost.ca/2012/04/24/employers...>

NLP EXAMPLE APPLICATIONS

Dear Ali,
let's meet tomorrow from 1:00-2:00pm in C5-202 to discuss
project status.
Yours,
Mohamed

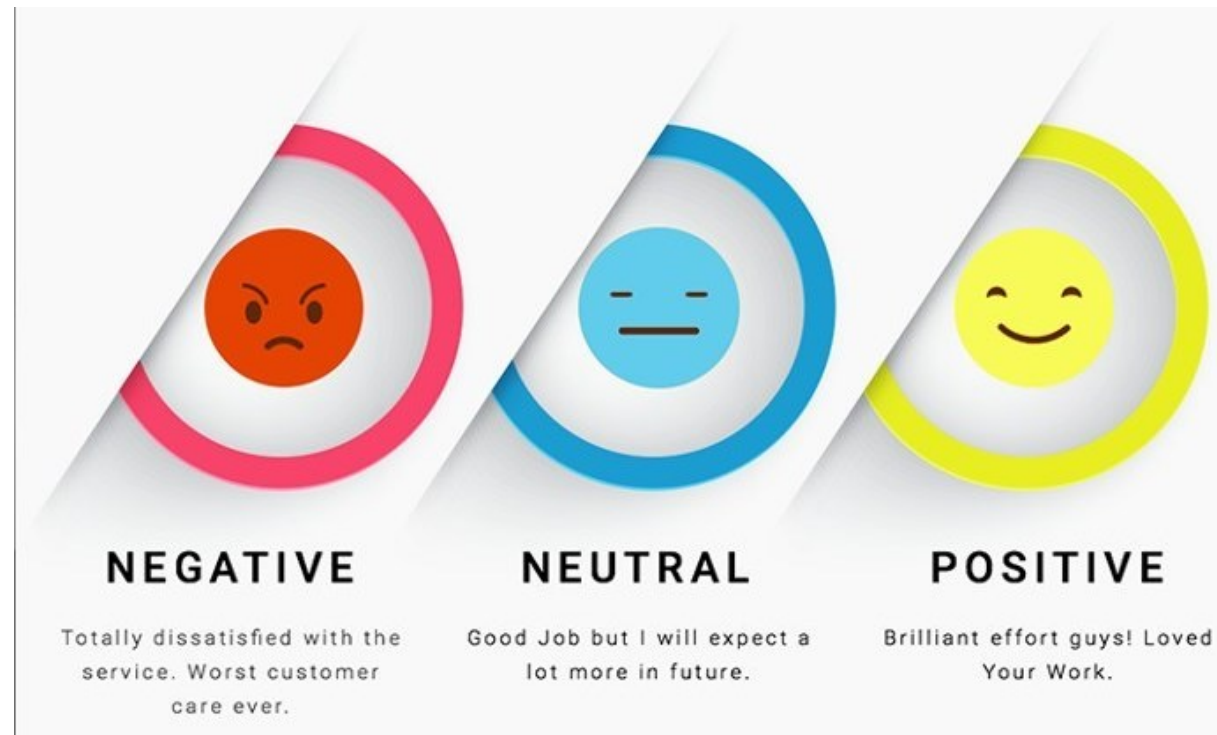
Information Extraction

Create Calendar entry

Event: project status meeting
Where: C5-202
Date: 15-Sept.-2017
Start: 1:00pm
End: 2:00pm

NLP EXAMPLE APPLICATIONS

Sentiment Analysis



NLP EXAMPLE APPLICATIONS

Text Mining

Discovering Periodic Patterns in Historical News

Fabon Dzogang, Thomas Lansdall-Welfare, FindMyPast Newspaper Team ✉, Nello Cristianini ✉

Published: November 8, 2016 • <http://dx.doi.org/10.1371/journal.pone.0165736>

Article	Authors	Metrics	Comments	Related Content
⌵				

Abstract

Introduction

Data Description

Methodology

Results and Discussion

Non Sinusoidal

Waveforms

Conclusions

Supporting Information

Acknowledgments

Author Contributions

References

Reader Comments (0)

Abstract

We address the problem of observing periodic changes in the behaviour of a large population, by analysing the daily contents of newspapers published in the United States and United Kingdom from 1836 to 1922. This is done by analysing the daily time series of the relative frequency of the 25K most frequent words for each country, resulting in the study of 50K time series for 31,755 days. Behaviours that are found to be strongly periodic include seasonal activities, such as hunting and harvesting. A strong connection with natural cycles is found, with a pronounced presence of fruits, vegetables, flowers and game. Periodicities dictated by religious or civil calendars are also detected and show a different wave-form than those provoked by weather. States that can be revealed include the presence of infectious disease, with clear annual peaks for fever, pneumonia and diarrhoea. Overall, 2% of the words are found to be strongly periodic, and the period most frequently found is 365 days. Comparisons between UK and US, and between modern and historical news, reveal how the fundamental cycles of life are shaped by the seasons, but also how this effect has been reduced in modern times.


481
View

7
Share

Download PDF ▾

Print

Share

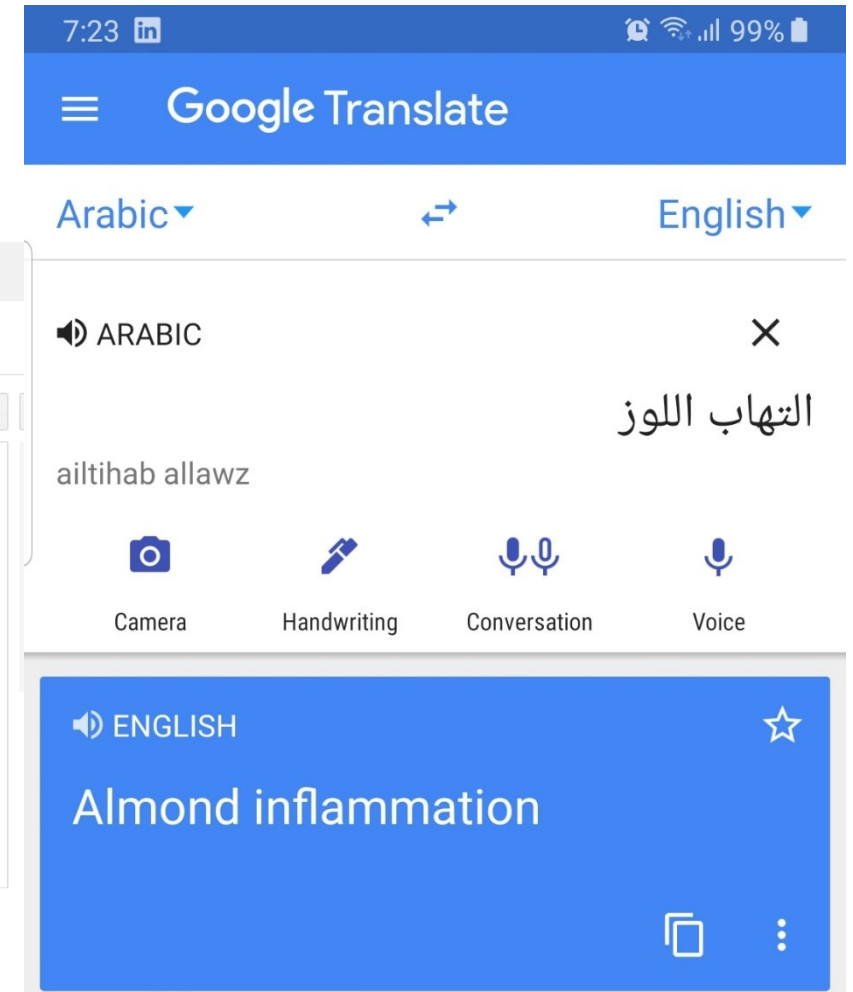
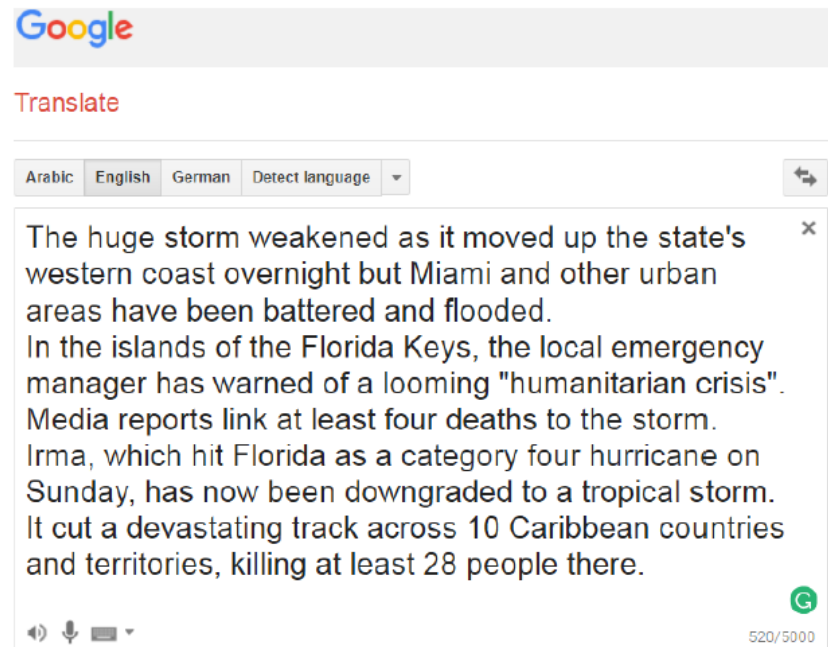
 CrossMark

Subject Areas

- Seasons
- Fourier analysis
- Summer
- Historical geography
- Elections
- Lectures
- Weather
- United Kingdom

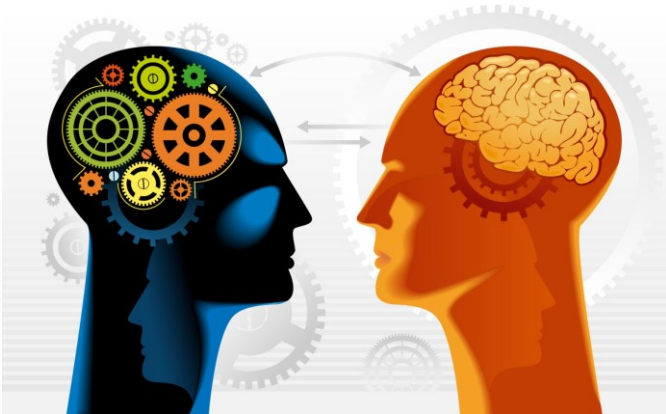
NLP EXAMPLE APPLICATIONS

Machine Translation



NLP EXAMPLE APPLICATIONS

Question Answering



Question: Who is the daughter of Bill Clinton married to?

Answer: Marc Mezvinsky

QA System

Knowledge Bases



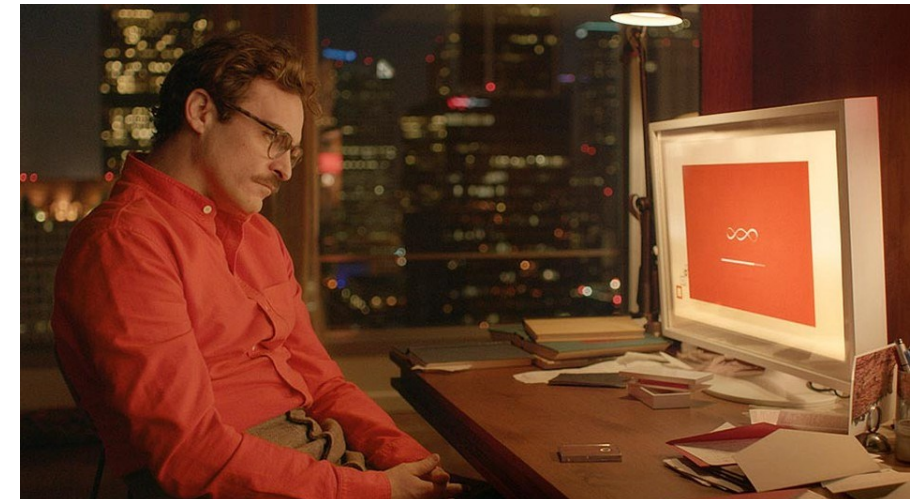
Datasets

NLP EXAMPLE APPLICATIONS

Conversational AI



ELIZA: What's on your mind today? Please let me know your opinion.
YOU: About family.
ELIZA: I see.
YOU: I think Sazae's family is good.
ELIZA: What caused you to feel that way?
YOU: Because the family members believe in themselves.
ELIZA: What other reasons might there be?
YOU: Well, I don't know.
ELIZA: I'd like to know more about your opinion.
YOU: I don't like my family so much.
ELIZA: That is quite interesting.
YOU: Yeah.
ELIZA: I'd like to know more about your opinion.
YOU: My family members aren't friendly, so I can't talk with them.
ELIZA: I see.
YOU: Do you have family?
ELIZA: Why do you think that?
YOU: I don't know who you are.
ELIZA: Why do you ask?
YOU: I'm sorry.
ELIZA: That's okay. Please continue.



NLP TECHNOLOGY

mostly solved

Spam detection

Let's go to Agra!



You won 1 BILLION USD ...



Part-of-speech (POS) tagging

ADJ ADJ NOUN VERB ADV

Colorless green ideas sleep furiously.

Named entity recognition (NER)

PERSON ORG LOC

Einstein met with UN officials in Princeton

making good progress

Sentiment analysis

Best roast chicken in San Francisco!



The waiter ignored us for 20 minutes.



Coreference resolution

Carter told Mubarak he shouldn't run again.

Word sense disambiguation (WSD)

I need new batteries for my *mouse*.



Machine translation (MT)

第13届上海国际电影节开幕...



The 13th Shanghai International Film Festival...

Information extraction (IE)

You're invited to our dinner party, Friday May 27 at 8:30



Party
May 27
add

still really hard

Question answering (QA)

Q. How effective is ibuprofen in reducing fever in patients with acute febrile illness?

Paraphrase

XYZ acquired ABC yesterday

ABC has been taken over by XYZ

Summarization

The Dow Jones is up

The S&P500 jumped

Housing prices rose



Economy is good

Dialog

Where is Citizen Kane playing in SF?



Castro Theatre at 7:30. Do you want a ticket?



**This slide is taken from Dan Jurafsky*

WHY IS NLP HARD?

Ambiguity ...at all levels of analysis

- Ambiguity is resolved with **knowledge**: world knowledge, contextual knowledge, statistical knowledge

Phonetics and phonology:

- Interpreting a speech signal: "I scream is delicious" vs. "ice cream is delicious"

Morphology: unionized = union + ized? un + ionized?

Syntax: Squad helps dog bite victim. (Does Squad help dog or victim?)

Semantics: Jack invited Mary to the ball. (ball could be “dance” or a “decoration ball”)

Discourse: Merck & Co. formed a joint venture with Ache Group, of Brazil. It will be called Prodome Ltd. (“it” refers to what?)

BUILDING A COMPUTER THAT 'UNDERSTANDS' TEXT: FUNDAMENTAL ELEMENTS OF THE NLP PIPELINE

**These slides are taken from Hongning Wang @ UVa*

TOKENIZATION/SEGMENTATION (BASIC)

Split text into words and sentences

- Task: what is the most **likely** segmentation /tokenization?

There was an earthquake near
D.C. I've even felt it in
Philadelphia, New York, etc.

There + was + an +
earthquake + near + D.C.

I + ve + even + felt + it + in +
Philadelphia, + New + York, +
etc.

Challenges – What to do about:

- **Sentence/word boundaries?**
- **Negation?**
- **Punctuations? Numbers?**
- **Shortened text?**
- **Punctuations that are integral part of names?**
- **Markup symbols in HTML? In social platforms?**
- **Tokenizing other languages?**

لا تدخل الآن، فالقاعة تعج بهم.

NORMALIZATION (BASIC)

Transform text into a single canonical form

- Task: what is the most **likely** normalization?

There were 3 résumés posted
to our recruitment system in
Dec. 5th. That is sooooo few!!

There + were + three + resumes + posted + to
+ our + recruitment + system + in + December
+ fifth + . + ▽ + That + is + so + few + !

لَا تَدْخُلِ الْآنَ، فَالْقَاعَةُ تَعْجُّ بِهِمْ.

Challenges – What to do about:

- **Non-standard text?**
- **Task-dependency?**
- **Multiple languages?**

STOPWORDS REMOVAL (BASIC)

Remove words that do not contribute to meaning

- Task: what is the most **likely** set of words that should be ignored?

Challenges – What to do about:

- **Negation and context?**
- **Desired dimensionality?**
- **Semantics?**

There was an earthquake near
D.C. I've even felt it in
Philadelphia, New York, etc.

There + was + an +
earthquake + **near** + D.C.

I + ve + even + felt + it + in +
Philadelphia, + **New** + York, +
etc.

لا تدخل الآن، فالقاعة تعج بهم.

STEMMING AND LEMMATIZATION (BASIC)

Reduce word to its root/lemma

- Task: what is the most **likely** root for a word?
- Suffix-stripping (or more general affix stripping) algorithms
- **Lemmatization** algorithms
- n-gram analysis

In our last meeting, we agreed
it is better that meeting the
client would be next week to
discuss requirements.

meet , **agree** , **good** , **meet** , **will** , **require**

تعتمد بلدان العالم على استخدام أنظمة الحاسب

Challenges – What to do about:

- **Affixes and semantics?**
- **Language?**

PART-OF-SPEECH TAGGING (MEDIUM)

Marking up a word in a text (corpus) as corresponding to a particular part of speech

- Task: what is the most **likely** tag sequence

Challenges – What to do about:

- **Ambiguity of meaning?**
- **Many-to-one tags?**
- **Language?**

A + dog + is + chasing + a + boy + on + the + playground

A + dog + is + chasing + a + boy + on + the + playground

Det Noun Aux Verb Det Noun Prep Det Noun

لا تدخل الآن، فالقاعة تعج بهم.

NAMED ENTITY RECOGNITION (ADVANCED)

Determine text mapping to proper names

- Task: what is the most **likely** mapping?

Challenges – What to do about:

- **Types of NEs?**
- **Annotation labor?**
- **Concept hierarchies?**
- **Scope and context?**
- **Language?**

Its initial Board of Visitors included U.S. Presidents Thomas Jefferson, James Madison, and James Monroe.

Its initial **Board of Visitors** included **U.S.** Presidents **Thomas Jefferson, James Madison, and James Monroe.**

Organization, Location, Person

SUMMARY: INITIAL STAGES OF TEXT PROCESSING

Tokenization

- Cut character sequence into word tokens
 - Consider white spaces, punctuation marks, hyphens, apostrophe, etc.
 - Deal with *“John’s”, a state-of-the-art solution*

Case Folding

- Reduce all letters to lower case. In other tasks like machine translation, case is important

Normalization

- Map text and query term to same form
 - You want **U.S.A.** and **USA** to match, but not **C.A.T.** and **cat**!

Stemming

- We may wish different forms of a root to match
 - *authorize, authorization*

Stop words

- We may omit very common words (or not)
 - *the, a, to, of*

IR BASICS – TERM-DOCUMENT INCIDENCE MATRICES

UNSTRUCTURED DATA IN 1620

Example query: Which plays of Shakespeare contain the words *Brutus* AND *Caesar* but NOT *Calpurnia*?

One could grep all of Shakespeare's plays for *Brutus* and *Caesar*, then strip out lines containing *Calpurnia*?

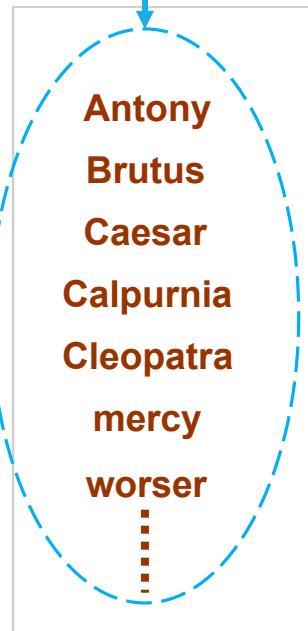
Why is that not the answer?

- Slow (for large corpora)
- NOT *Calpurnia* is non-trivial
- Other operations (e.g., find the word *Romans* near *countrymen*) not feasible
- Ranked retrieval (best documents to return)

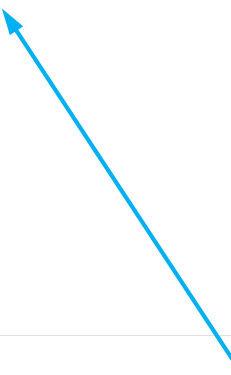
No. of Shakespeare plays: **37**
Avg. no. of words per play: **22,000**

TERM-DOCUMENT INCIDENCE MATRIX

All distinct words
(terms) in **all plays**



	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: Brutus AND Caesar
BUT NOT Calpurnia**

1 if document contains
word, 0 otherwise

INCIDENCE VECTORS

So we have a 0/1 **vector** for each term

To answer query: take the vectors for *Brutus*, *Caesar* and *Calpurnia* (**complemented**)
→ bitwise AND

- 110100 AND

- 110111 AND

- 101111 =

- **100100**

	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

ANSWERS TO QUERY

Antony and Cleopatra, Act III, Scene ii

Agrippa [Aside to DOMITIUS ENOBARBUS]: Why, Enobarbus,
When Antony found Julius *Caesar* dead,
He cried almost to roaring; and he wept
When at Philippi he found *Brutus* slain.

Hamlet, Act III, Scene ii

Lord Polonius: I did enact Julius *Caesar* I was killed i' the
Capitol; *Brutus* killed me.

BIGGER COLLECTIONS

Consider $N = 1$ million documents, each with about 1000 words

Avg 6 bytes/word including spaces/punctuation

- 6GB of data in the documents

Say there are $M = 500K$ **distinct terms** among these

CAN'T BUILD THE MATRIX

500K x 1M matrix has half-a-trillion 0's and 1's

But it has no more than one billion 1's



- matrix is extremely sparse

What's a better representation?

- We only record the 1 positions

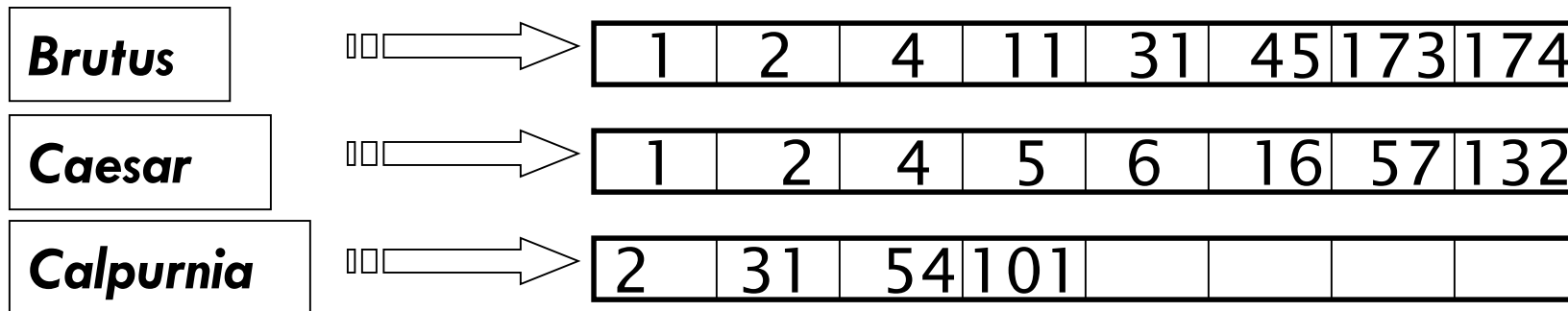
IR BASICS – THE INVERTED INDEX

INVERTED INDEX

For each term t , we must store a list of all documents that contain t

Identify each document by a *docID*, a document serial number

Can we use fixed-size arrays for this?

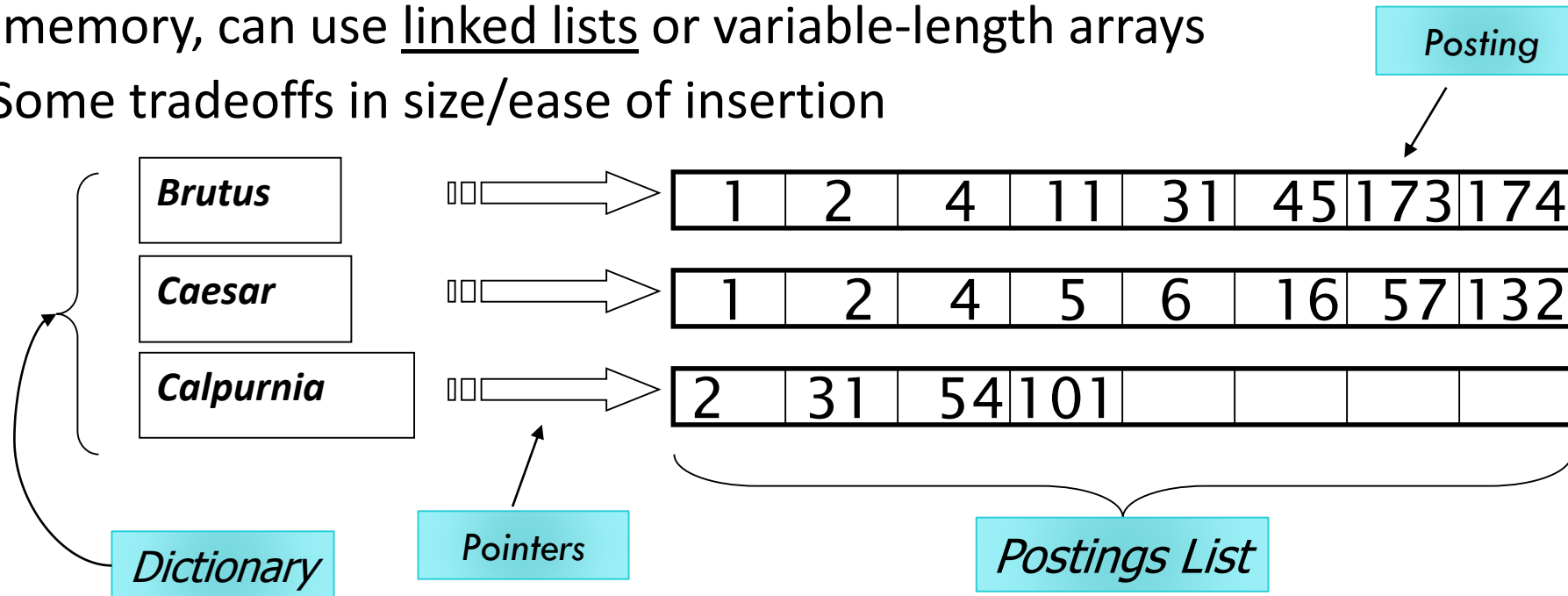


What happens if the word **Caesar** is added to document 14?

INVERTED INDEX

We need variable-size **postings lists**

- On disk, a continuous run of postings is normal and best
- In memory, can use linked lists or variable-length arrays
 - Some tradeoffs in size/ease of insertion



Sorted by *docID* (more later on why)

INVERTED INDEX CONSTRUCTION

Documents collection



Tokenizer

Token stream

Friends, Romans, countrymen.

⋮

Friends

Romans

Countrymen

Linguistic modules

Case folding, Normalization,
stemming, stop words removal, etc.

Modified tokens

friend

roman

countryman

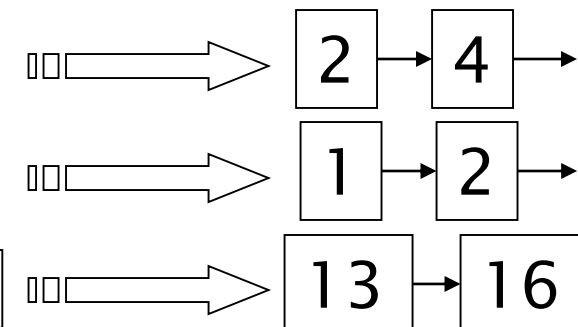
Indexer

Inverted index

friend

roman

countryman



INDEXER STEPS: **TOKEN SEQUENCE**

Sequence of (**Modified** token, Document ID) pairs

Doc 1

I did enact Julius
Caesar I was killed
i' the Capitol;
Brutus killed me.

Doc 2

So let it be with
Caesar. The noble
Brutus hath told you
Caesar was ambitious



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

INDEXER STEPS: SORT

Sort by terms

- And then *docID*

Core indexing step

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



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

INDEXER STEPS: **DICTIONARY & POSTINGS**

- Multiple term entries in a single document are merged
- Split into Dictionary and Postings
- Doc. frequency** information is added

Why frequency?
Will discuss later

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



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
i	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Inverted index

HOW DO WE RESPOND TO A QUERY?

How do we process a query?

- What kinds of queries can we process?

Queries containing **one term** are pretty easy

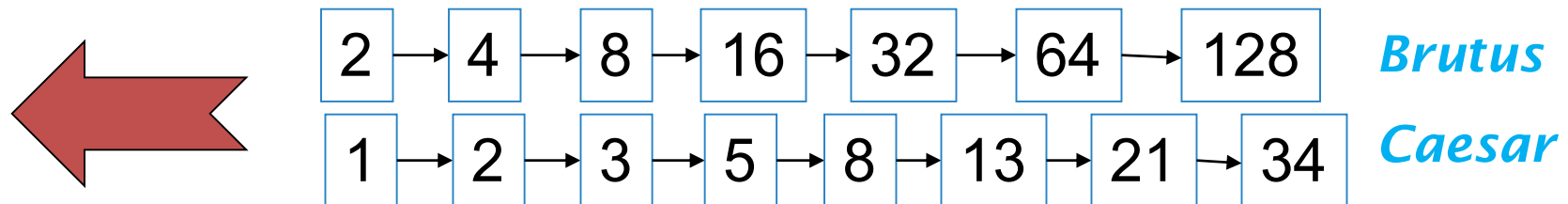
- Just get all postings linked to the term in the dictionary

QUERY PROCESSING: AND

Consider processing the query:

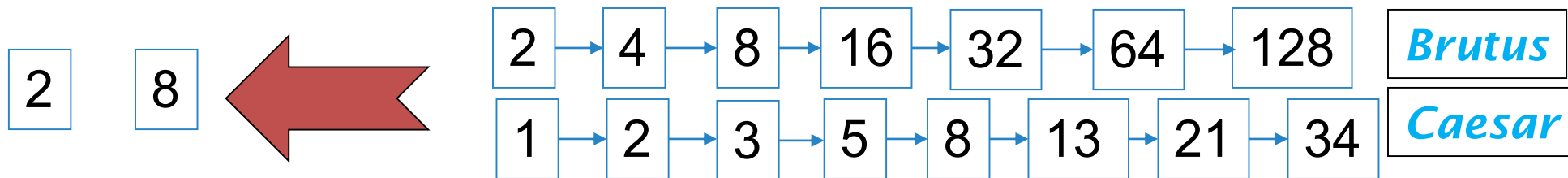
Brutus AND Caesar

- Locate *Brutus* in the Dictionary;
 - Retrieve its postings
- Locate *Caesar* in the Dictionary;
 - Retrieve its postings
- “Merge” the two postings (intersect the document sets):



THE MERGE

Walk through the two postings simultaneously, in time linear in the total number of postings entries



If the list lengths are x and y , the merge takes $O(x + y)$ operations

Crucial: postings sorted by *docID*

INTERSECTING TWO POSTINGS LISTS (A “MERGE” ALGORITHM)

```
INTERSECT( $p_1, p_2$ )
1   $answer \leftarrow \langle \rangle$ 
2  while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3  do if  $docID(p_1) = docID(p_2)$ 
4      then  $\text{ADD}(answer, docID(p_1))$ 
5           $p_1 \leftarrow next(p_1)$ 
6           $p_2 \leftarrow next(p_2)$ 
7      else if  $docID(p_1) < docID(p_2)$ 
8          then  $p_1 \leftarrow next(p_1)$ 
9          else  $p_2 \leftarrow next(p_2)$ 
10 return  $answer$ 
```

QUERY OPTIMIZATION: MORE ANDS!

Consider a query that is an AND of n **terms**, $n > 2$

- For each of the terms, get its postings list, then AND them together

Example query: *Brutus* AND *Calpurnia* AND *Caesar*

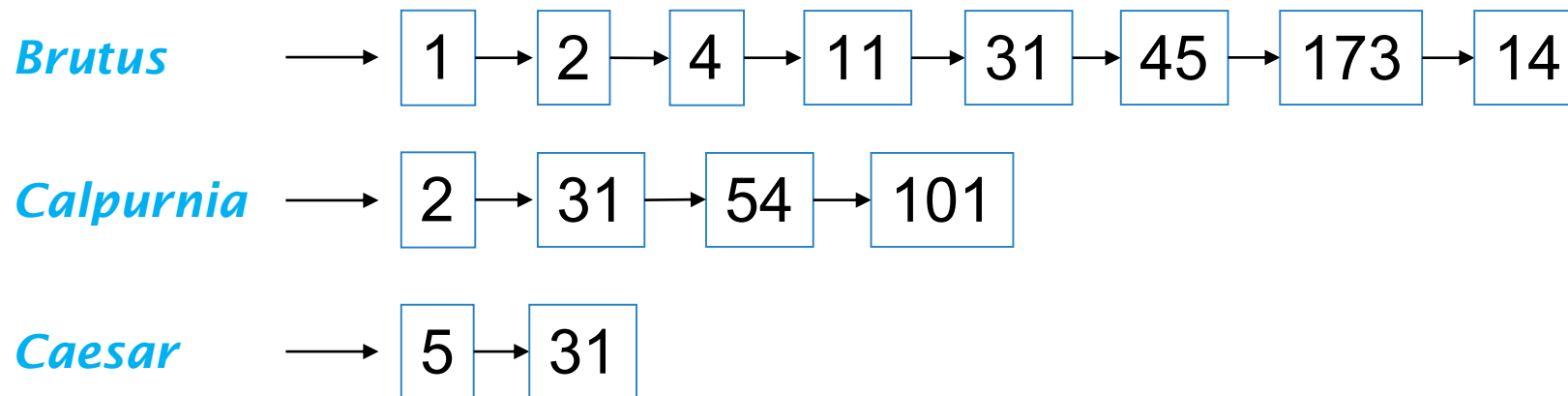
- What is the best order for processing this query?

QUERY OPTIMIZATION: MORE ANDS!

Example query: *Brutus* AND *Calpurnia* AND *Caesar*

Simple and effective optimization: Process in order of increasing frequency

- Start with the shortest postings list, then keep cutting further
- In this example, first *Caesar*, then *Calpurnia*, then *Brutus*



OPTIMIZED INTERSECTION ALGORITHM FOR CONJUNCTIVE QUERIES

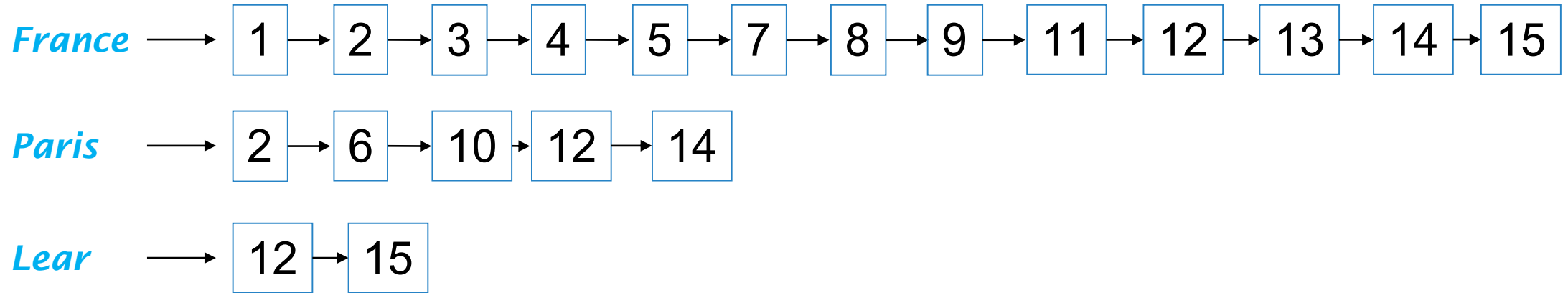
```
INTERSECT( $\langle t_1, \dots, t_n \rangle$ )  
1  terms  $\leftarrow$  SORTBYINCREASINGFREQUENCY( $\langle t_1, \dots, t_n \rangle$ )  
2  result  $\leftarrow$  postings(first(terms))  
3  terms  $\leftarrow$  rest(terms)  
4  while terms  $\neq$  NIL and result  $\neq$  NIL  
5  do result  $\leftarrow$  INTERSECT(result, postings(first(terms)))  
6    terms  $\leftarrow$  rest(terms)  
7  return result
```

MORE GENERAL OPTIMIZATION

Example query: (*madding* or *crowd*) and (*ignoble* or *strife*)

- Get frequencies for all terms
- Estimate the size of each or by the sum of its frequencies (conservative)
- Process in increasing order of or sizes

EXERCISE



Compute hit list for ((*paris* AND NOT *france*) OR *lear*)

NEXT TIME

Phrase Queries

Vector Space Model

Term Weighting

REFERENCES

This lecture is heavily relying on the following courses:

- CS 276 / LING 286: Information Retrieval and Web Search, Stanford University
- Natural Language Processing Lecture Slides from the Stanford Coursera course by Dan Jurafsky and Christopher Manning