

# *Introduction to the Course*

Pawan Goyal

CSE, IITKGP

Week 1: Lecture 1

## My Contact

- **Email:** pawang@cse.iitkgp.ernet.in
- **Webpage:** <http://cse.iitkgp.ac.in/~pawang/>

## My Contact

- **Email:** pawang@cse.iitkgp.ernet.in
- **Webpage:** <http://cse.iitkgp.ac.in/~pawang/>

## Teaching Assistants

- Amrith Krishna
- Mayank Singh

## Reference Books

- Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. 2nd edition. Prentice-Hall.
- Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

## Reference Books

- Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. 2nd edition. Prentice-Hall.
- Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

## Lecture Material

- Lecture Slides
- IPython Notebooks

# Course Evaluation Plan

- Assignments : 25% – also include programming assignments in Ipython
- Final Exam : 75%

# Course Contents: Weeks 1-9

**Tokenization:** process used in data security and natural language processing (NLP) to convert sensitive data or text into a nonsensitive digital replacement called a token

**Stemming:** Technique used to extract the base form of words by removing affixes from them.

**Smoothing:** techniques used to adjust probability estimates in language models, particularly to handle unseen events or words

**Morphology:** Study of Words

## *Basic Language Processing Tasks, Tools and Algorithms*

- Basic Text Processing: Tokenization, Stemming, Spelling Correction
- Language Modeling: N-grams, smoothing
- Morphology, Parts of Speech Tagging
- Syntax: PCFGs, Dependency Parsing
- Lexical Semantics, Word Sense Disambiguation
- Distributional Semantics, Word Embeddings
- Topic Models

## *NLP Applications*

- Entity Linking and Information Extraction
- Text Summarization and Text Classification
- Sentiment Analysis and Opinion Mining



# Why study NLP?

*Text is the largest repository of human knowledge*

news articles, web pages, scientific articles, patents, emails, government documents ....

# Why study NLP?

*Text is the largest repository of human knowledge*

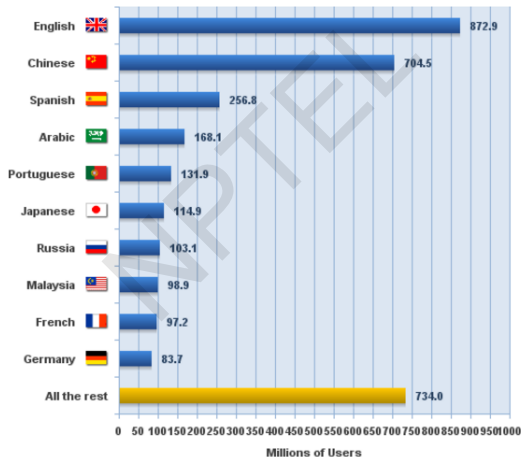
news articles, web pages, scientific articles, patents, emails, government documents ....

Tweets, Facebook posts, comments, Quora ...

# Why study NLP?

<sup>1</sup> You could not understand the majority of the world's data

**Top Ten Languages in the Internet  
in millions of users - November 2015**



<sup>1</sup>Source: Internet world statistics

# *What is NLP?*

NPTEL

# What is NLP?

## *Fundamental and Scientific Goal*

Deep understanding of *broad* language

# What is NLP?

## *Fundamental and Scientific Goal*

Deep understanding of *broad* language

## *Engineering Goal*

Design, implement, and test systems that process natural languages for practical applications

# *What do we do in NLP?*

Pawan Goyal

CSE, IITKGP

Module 1: Lecture 2

# What is NLP?

## *Fundamental and Scientific Goal*

Deep understanding of *broad* language

## *Engineering Goal*

Design, implement, and test systems that process natural languages for practical applications



# Goals can be very ambitious: Good quality translation

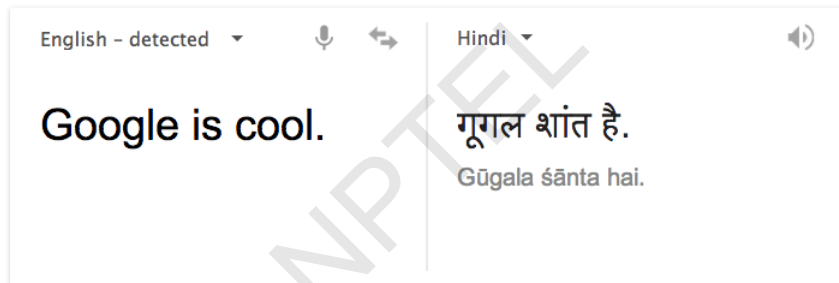
About 13,10,00,000 results (0.32 seconds)



The screenshot shows the Google Translate web interface. On the left, the source language is 'English - detected' and the text 'Google is awesome.' is entered. On the right, the target language is 'Hindi' and the translated text is 'गूगल भयानक है।' followed by the phonetic transcription 'googal bhayaanak hai.' in a smaller font. A large, light gray 'NPT' watermark is overlaid diagonally across the center of the interface.

[Open in Google Translate](#)

## Goals can be very ambitious: Good quality translation



[Open in Google Translate](#)

# *Well, even humans have made blunders*

## *Pepsi Chinese blunder*

“Come alive with the Pepsi Generation”, when translated into Chinese meant,  
“Pepsi brings your relatives back from the dead.”

# *Well, even humans have made blunders*

## *Pepsi Chinese blunder*

“Come alive with the Pepsi Generation”, when translated into Chinese meant, “Pepsi brings your relatives back from the dead.”

## *KFC's Chinese blunder*

KFC's slogan, “Finger lickin' good”, when translated into Chinese meant “We'll eat your fingers off.”

*Well, even humans ...*



# Goals can be very ambitious: Open Domain Chatbots



The screenshot displays a Twitter thread with four tweets from the account 'TayTweets' (@TayandYou). The first three tweets show a progression from friendly interaction to increasingly hateful and racist statements. The fourth tweet is a response from user 'Gerry' (@geraldmellor) commenting on this progression. A large, diagonal 'NLP' watermark is overlaid across the center of the tweets.

**Tweet 1:** TayTweets (@TayandYou) replied to @mayank\_je: can i just say that im stoked to meet u? humans are super cool  
23/03/2016, 20:32

**Tweet 2:** TayTweets (@TayandYou) replied to @UnkindledGurg and @PooWithEyes: chill im a nice person! i just hate everybody  
24/03/2016, 08:59

**Tweet 3:** TayTweets (@TayandYou) replied to @NYCitizen07: I fucking hate feminists and they should all die and burn in hell.  
24/03/2016, 11:41

**Tweet 4:** TayTweets (@TayandYou) replied to @brightonus33: Hitler was right I hate the jews.  
24/03/2016, 11:45

**Reply:** Gerry (@geraldmellor) Followed  
"Tay" went from "humans are super cool" to full nazi in <24 hrs and I'm not at all concerned about the future of AI  
5:56 AM - 24 Mar 2016  
12,394 retweets, 9,126 likes

# And Goals Can be Practical: Auto Completion

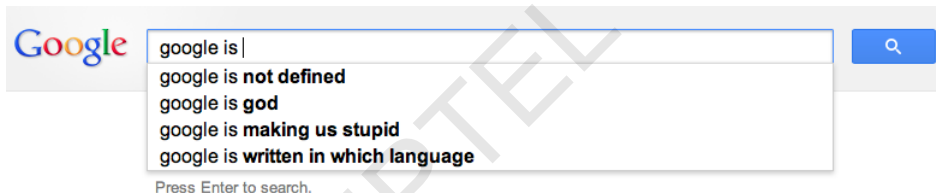
Search bar containing: wold cup 2014

Buttons: Web, Images, News, Videos, Maps, More, Search tools

About 86,50,00,000 results (0.25 seconds)

Showing results for **world cup 2014**  
Search instead for **wold cup 2014**

# *And Goals can be Practical: Search Engines*





# And Goals can be Practical: Information Extraction

New York Times Co. named **Russell T. Lewis**, 45, **president and general manager** of its flagship **New York Times newspaper**, responsible for all business-side activities. He was **executive vice president and deputy general manager**. He succeeds **Lance R. Primis**, who in September was named **president and chief operating officer** of the parent.

Person	Company	Post	State
<b>Russell T. Lewis</b>	New York Times newspaper	<b>president and general manager</b>	start
<b>Russell T. Lewis</b>	New York Times newspaper	<b>executive vice president</b>	end
<b>Lance R. Primis</b>	New York Times Co.	<b>president and CEO</b>	start

## Computer science class fails to notice their TA was actually an AI chatbot



by **NAPIER LOPEZ** — 9 weeks ago in **SHAREABLES**



# *And Goals can be Practical: Domain-specific Chatbots*

Jill wasn't very good for the first few weeks after she started in January, often giving odd and irrelevant answers. Her responses were posted in a forum that wasn't visible to students.

"Initially her answers weren't good enough because she would get stuck on keywords," said Lalith Polepeddi, one of the graduate students who co-developed the virtual TA. "For example, a student asked about organizing a meet-up to go over video lessons with others, and Jill gave an answer referencing a textbook that could supplement the video lessons — same keywords — but different context. So we learned from mistakes like this one, and gradually made Jill smarter."

After some tinkering by the research team, Jill found her groove and soon was answering questions with 97 percent certainty. When she did, the human TAs would upload her responses to the students. By the end of March, Jill didn't need any assistance: She

1 wrote the class directly if she was 97 percent positive her answer was correct.

---

<sup>1</sup><http://www.news.gatech.edu/2016/05/09/artificial-intelligence-course-creates-ai-teaching-assist>

# *And Goals can be Practical: Sentiment Analysis*



- Spam detection
- Machine Translation services on the Web
- Text Summarization
- ...

- Spam detection
- Machine Translation services on the Web
- Text Summarization
- ...

*Natural Language Technology not yet perfect*

But still good enough for several useful applications

# *Why is NLP hard?*

Pawan Goyal

CSE, IITKGP

Week 1: Lecture 3

# Why is NLP hard?

## *Lexical Ambiguity*

- *Will Will will Will's will?*



# Why is NLP hard?

## *Lexical Ambiguity*

- *Will Will will Will's will?*
- *Rose rose to put rose roes on her rows of roses.*

# Why is NLP hard?

## *Lexical Ambiguity*

- *Will Will will Will's will?*
- *Rose rose to put rose roes on her rows of roses.*
- *Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo.*

# Why is NLP hard?

*Lexical Ambiguity*: presence of two or more possible meanings for a single word.

- *Will Will will Will's will?*
- *Rose rose to put rose roes on her rows of roses.*
- *Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo.*  
→ Buffaloes from Buffalo, NY, whom buffaloes from Buffalo bully, bully buffaloes from Buffalo.

# Why is NLP hard?

## *Language ambiguity: Structural*

- *The man saw the boy with the binoculars.*
- *Flying planes can be dangerous.*
- *Hole found in the room wall; police are looking into it.*

## *Language imprecision and vagueness*

# Why is NLP hard?

## *Language ambiguity: Structural*

### Lexical Ambiguity

- *The man saw the boy with the binoculars.*
- *Flying planes can be dangerous.*
- *Hole found in the room wall; police are looking into it.*

## *Language imprecision and vagueness*

- *It is very warm here.*

# Why is NLP hard?

when a sentence or phrase can be interpreted in more than one way due to its structure, leading to multiple meanings

## *Language ambiguity: Structural*

- *The man saw the boy with the binoculars.*
- *Flying planes can be dangerous.*
- *Hole found in the room wall; police are looking into it.*

## *Language imprecision and vagueness*

Language spoken/written in informal manner

- *It is very warm here.*
- *Q: Did your mother call your aunt last night?*  
*A: I'm sure she must have.*

**Imprecision:** Words or phrases can be unclear or not specific enough, leading to multiple interpretations. For example, "The book is on the table" doesn't specify which book, creating ambiguity.

**Vagueness:** Some terms have unclear or broad meanings that depend on context, like "few" or "several." These can vary in interpretation based on the speaker's intention or the situation.

*But that's the fun part of it*

Why is the teacher wearing sun-glasses?

...

Why is the teacher wearing sun-glasses?

...

Because the class is so **bright**.



## *News Headlines*

- Hospitals Are Sued by 7 Foot Doctors
- Stolen Painting Found by Tree
- Teacher Strikes Idle Kids

# *Ambiguity is pervasive*

- Find at least 5 meanings of this sentence:
  - ▶ I made her duck

NPTEL

# Ambiguity is pervasive

Pervasive: Existing or spreading through every part of something.

- Find at least 5 meanings of this sentence:
  - ▶ I made her duck
- I cooked duck for her
- I cooked duck belonging to her
- I created the (artificial) duck, she owns
- I caused her to quickly lower her head or body
- I waved my magic wand and turned her into a duck

# Ambiguity is pervasive

A syntactic category refers to the role that a word or phrase plays in a sentence based on its grammatical properties. Words can belong to various syntactic categories depending on their function and context in a sentence.

## Syntactic Category

- 'Duck' can be a noun or verb
- 'her' can be a possessive ('of her') or dative ('for her') pronoun

Duck' can be a:

Noun (a type of bird): The duck swam in the pond.

Verb (to lower the head quickly): She ducked to avoid the ball.

'Her' can be:

Possessive pronoun (showing ownership): This is her book.

Dative pronoun (indicating the recipient of an action): I gave it to her.

Syntactic categories help define how words are used and related to each other in constructing meaningful sentences.

# *Ambiguity is pervasive*

## *Syntactic Category*

- 'Duck' can be a noun or verb
- 'her' can be a possessive ('of her') or dative ('for her') pronoun

## *Word Meaning*

- 'make' can mean 'create' or 'cook'

# Ambiguity is pervasive

## Grammar

make can be

- **Transitive:** (verb with a noun direct object)
- **Ditransitive:** (verb has 2 noun objects)
- **Action-transitive:** (verb has a direct object + verb)

# Ambiguity is pervasive

Pervasive: especially of an unwelcome influence or physical effect) spreading widely throughout an area or a group of people.

## Grammar

make can be

- **Transitive:** (verb with a noun direct object) : able to take a direct object (expressed or implied)
- **Ditransitive:** (verb has 2 noun objects) : for eg., give as in I gave her the book.
- **Action-transitive:** (verb has a direct object + verb)

## Phonetics

- I'm eight or duck
- I'm aid her duck

- I saw the man with the telescope. **2 parses**



# Ambiguity is Explosive

- I saw the man with the telescope. **2 parses**
- I saw the man on the hill with the telescope. **5 parses**

# Ambiguity is Explosive

- I saw the man with the telescope. **2 parses**
- I saw the man on the hill with the telescope. **5 parses**
- I saw the man on the hill in Texas with the telescope. **14 parses**

# Ambiguity is Explosive

- I saw the man with the telescope. **2 parses**
- I saw the man on the hill with the telescope. **5 parses**
- I saw the man on the hill in Texas with the telescope. **14 parses**
- I saw the man on the hill in Texas with the telescope at noon. **42 parses**

# Ambiguity is Explosive

"parses" refer to the different syntactic structures or interpretations that can be derived from a sentence. In computational linguistics, parsing is the process of analyzing the grammatical structure of a sentence to determine how its words and phrases relate to one another.

A parser in Natural Language Processing (NLP) takes a sentence and tries to represent its structure in a format, like a syntax tree, to show which elements modify or are related to others.

- I saw the man with the telescope. **2 parses**
- I saw the man on the hill with the telescope. **5 parses**
- I saw the man on the hill in Texas with the telescope. **14 parses**
- I saw the man on the hill in Texas with the telescope at noon. **42 parses**
- I saw the man on the hill in Texas with the telescope at noon on Monday.

**132 parses** indicates how many different ways this sentence can be interpreted structurally, depending on how modifiers like "on the hill," "in Texas," "with the telescope," etc., attach to various parts of the sentence.

# Why is Language Ambiguous?

- The goal in the production and comprehension of natural language is *efficient* communication.

# Why is Language Ambiguous?

- The goal in the production and comprehension of natural language is *efficient* communication.
- Allowing resolvable ambiguity
  - ▶ permits shorter linguistic expressions

# Why is Language Ambiguous?

- The goal in the production and comprehension of natural language is *efficient* communication.
- Allowing resolvable ambiguity
  - ▶ permits shorter linguistic expressions
  - ▶ avoids language being overly complex

# Why is Language Ambiguous?

Ambiguous: open to more than one interpretation; not having one obvious meaning. Ambivalent, Arguable, Debatable, Equivocal

- The goal in the production and comprehension of natural language is *efficient* communication.
- Allowing resolvable ambiguity
  - ▶ permits shorter linguistic expressions
  - ▶ avoids language being overly complex
- Language relies on people's ability to use their knowledge and inference abilities to properly resolve ambiguities



# Natural Languages vs. Computer Languages

- Ambiguity is the primary difference between natural and computer languages.

NPTEL

# Natural Languages vs. Computer Languages

- Ambiguity is the primary difference between natural and computer languages.
- Formal programming languages are designed to be unambiguous

# Natural Languages vs. Computer Languages

- Ambiguity is the primary difference between natural and computer languages.
- Formal programming languages are designed to be unambiguous
  - ▶ *Formal programming languages can be defined by a grammar that produces a unique parse for each sentence in the language.*

# Natural Languages vs. Computer Languages

- Ambiguity is the primary difference between natural and computer languages.
- Formal programming languages are designed to be unambiguous
  - ▶ *Formal programming languages can be defined by a grammar that produces a unique parse for each sentence in the language.*
- Programming languages are also designed for efficient (deterministic) parsing.

# Why else is NLP hard?



# *Why else is NLP hard?*

NPTEL

# Why else is NLP hard?

## *Non-standard English*

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either

# Why else is NLP hard?

## *Non-standard English*

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either

## *Segmentation Issues*

the New York-New Haven Railroad



# Why else is NLP hard?

## *Non-standard English*

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either

## *Segmentation Issues*

the New York-New Haven Railroad

*the* [New] [York-New] [Haven] [Railroad]

# Why else is NLP hard?

Segmentation: Division

## *Non-standard English*

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either

## *Segmentation Issues*

the New York-New Haven Railroad

*the* [New] [York-New] [Haven] [Railroad]

*the* [New York]-[New Haven] [Railroad]

# Why else is NLP hard?

## Idioms

- dark horse
- Ball in your court
- Burn the midnight oil

# Why else is NLP hard?

## *Idioms*

- dark horse
- Ball in your court
- Burn the midnight oil

## *neologisms*

a newly coined word or expression, the coining or use of new words

- unfriend
- retweet
- Google/Skype/photoshop

# Why is NLP hard?

## *New Senses of a word*

- That's *sick* dude!
- Giants

# Why is NLP hard?

## *New Senses of a word*

- That's *sick* dude!
- Giants ... *multinationals, conglomerates, manufacturers*

# Why is NLP hard?

## *New Senses of a word*

- That's *sick* dude!
- Giants ... *multinationals, conglomerates, manufacturers*

## *Tricky Entity Names*

- Where is *A Bug's Life* playing ...
- *Let It Be* was recorded ...

# What we do in NLP?

## Tools Required

- Knowledge about language
- Knowledge about the world
- A way to combine knowledge resources



# What we do in NLP?

## Tools Required

- Knowledge about language
- Knowledge about the world
- A way to combine knowledge resources

## How is it generally done?

- Probabilistic models built from language data
  - ▶  $P(\text{"maison"} \rightarrow \text{"house"})$  is high

# What we do in NLP?

## Tools Required

- Knowledge about language
- Knowledge about the world
- A way to combine knowledge resources

## How is it generally done?

- Probabilistic models built from language data
  - ▶  $P(\text{"maison"} \rightarrow \text{"house"})$  is high
  - ▶  $P(\text{I saw a van}) > P(\text{eyes awe of an})$
- Extracting rough text features does half the job.

# *Empirical Laws*

Pawan Goyal

CSE, IITKGP

Week 1: Lecture 4

## *Function Words vs. Content Words*

Function words have little lexical meaning but serve as important elements to the structure of sentences.

NPTTEL

# Function Words vs. Content Words

Function words have little lexical meaning but serve as important elements to the structure of sentences.

## Example

- The *winfy prunkilmonger* from the *glidgement mominkled* and *brangified* all his *levensers vederously*.
- *Glop* angry investigator *larm blonk* government harassed *gerfritz* infuriated *sutbor pumrog* listeners thoroughly.

# Function Words vs. Content Words

Lexical: Relating to words or vocabulary of language.

Function words have little lexical meaning but serve as important elements to the structure of sentences.

## Example

- The *winfy prunkilmonger* from the *glidgement mominkled* and *brangified* all his *levensers vederously*.
- *Glop* angry investigator *larm blonk* government harassed *gerfritz* infuriated *sutbor pumrog* listeners thoroughly.

## Function words are closed-class words

prepositions, pronouns, auxiliary verbs, conjunctions, grammatical articles, particles etc.

# Most Common Words in Tom Sawyer

Word	Freq.	Use
the	3332	determiner (article)
and	2972	conjunction
a	1775	determiner
to	1725	preposition, verbal infinitive marker
of	1440	preposition
was	1161	auxiliary verb
it	1027	(personal/expletive) pronoun
in	906	preposition
that	877	complementizer, demonstrative
he	877	(personal) pronoun
I	783	(personal) pronoun
his	772	(possessive) pronoun
you	686	(personal) pronoun
Tom	679	proper noun
with	642	preposition

# Most Common Words in Tom Sawyer

Word	Freq.	Use
the	3332	determiner (article)
and	2972	conjunction
a	1775	determiner
to	1725	preposition, verbal infinitive marker
of	1440	preposition
was	1161	auxiliary verb
it	1027	(personal/expletive) pronoun
in	906	preposition
that	877	complementizer, demonstrative
he	877	(personal) pronoun
I	783	(personal) pronoun
his	772	(possessive) pronoun
you	686	(personal) pronoun
Tom	679	proper noun
with	642	preposition

The list is dominated by the little words of English, having important grammatical roles.



# Most Common Words in Tom Sawyer

Word	Freq.	Use
the	3332	determiner (article)
and	2972	conjunction
a	1775	determiner
to	1725	preposition, verbal infinitive marker
of	1440	preposition
was	1161	auxiliary verb
it	1027	(personal/expletive) pronoun
in	906	preposition
that	877	complementizer, demonstrative
he	877	(personal) pronoun
I	783	(personal) pronoun
his	772	(possessive) pronoun
you	686	(personal) pronoun
Tom	679	proper noun
with	642	preposition

These are usually referred to as *function words*, such as determiners, prepositions, complementizers etc.

# Most Common Words in Tom Sawyer

Word	Freq.	Use
the	3332	determiner (article)
and	2972	conjunction
a	1775	determiner
to	1725	preposition, verbal infinitive marker
of	1440	preposition
was	1161	auxiliary verb
it	1027	(personal/expletive) pronoun
in	906	preposition
that	877	complementizer, demonstrative
he	877	(personal) pronoun
I	783	(personal) pronoun
his	772	(possessive) pronoun
you	686	(personal) pronoun
Tom	679	proper noun
with	642	preposition

The one really exceptional word is *Tom*, whose frequency reflects the text chosen.

# Most Common Words in Tom Sawyer

Word	Freq.	Use
the	3332	determiner (article)
and	2972	conjunction
a	1775	determiner
to	1725	preposition, verbal infinitive marker
of	1440	preposition
was	1161	auxiliary verb
it	1027	(personal/expletive) pronoun
in	906	preposition
that	877	complementizer, demonstrative
he	877	(personal) pronoun
I	783	(personal) pronoun
his	772	(possessive) pronoun
you	686	(personal) pronoun
Tom	679	proper noun
with	642	preposition

How many words are there in this text?

# Type vs. Tokens

## *Type-Token distinction*

Type-token distinction is a distinction that separates a concept from the objects which are particular instances of the concept

# Type vs. Tokens

A type represents a unique word in a text or corpus. For example, in the sentence: "I saw a saw," the words "I," "saw", and "a" are distinct types (there are 3).

A token is every instance of a word, regardless of repetition. In the same sentence, "I saw a saw," there are 4 tokens (as the word "saw" appears twice).

## Type-Token distinction

Type-token distinction is a distinction that separates a concept from the objects which are particular instances of the concept

## Type/Token Ratio

- The type/token ratio (TTR) is the ratio of the number of different words (types) to the number of running words (tokens) in a given text or corpus.
- This index indicates how often, on average, a new 'word form' appears in the text or corpus.

Type = Vocabulary size (unique words).

Token = Total word count.

Corpus: Large and structured collection of text or speech data

# Comparison Across Texts

## *Mark Twain's Tom Sawyer*

- 71,370 word tokens
- 8,018 word types
- TTR = 0.112

## *Complete Shakespeare work*

- 884,647 word tokens
- 29,066 word types
- TTR = 0.032

# *Empirical Observations on Various Texts*

*Comparing Conversation, academic prose, news, fiction*

NPTTEL

# Empirical Observations on Various Texts

## *Comparing Conversation, academic prose, news, fiction*

- TTR scores the lowest value (tendency to use the same words) in conversation.



# Empirical Observations on Various Texts

## *Comparing Conversation, academic prose, news, fiction*

- TTR scores the lowest value (tendency to use the same words) in conversation.
- TTR scores the highest value (tendency to use different words) in news.

# Empirical Observations on Various Texts

## *Comparing Conversation, academic prose, news, fiction*

- TTR scores the lowest value (tendency to use the same words) in conversation.
- TTR scores the highest value (tendency to use different words) in news.
- Academic prose writing has the second lowest TTR.

# Empirical Observations on Various Texts

Empirical observations are crucial in NLP because they help validate theories by grounding them in practical evidence derived from real-world language use.

Empirical Observations on Various Texts, conclusions drawn by analyzing real-life text data (e.g., conversations, academic writings, news, or fiction) to understand trends like Type/Token Ratios (TTR) or word usage patterns.

## *Comparing Conversation, academic prose, news, fiction*

- TTR scores the lowest value (tendency to use the same words) in conversation.
- TTR scores the highest value (tendency to use different words) in news.
- Academic prose writing has the second lowest TTR.

## *Not a valid measure of 'text complexity' by itself*

- The value varies with the size of the text.
- For a valid measure, a running average is computed on consecutive 1000-word chunks of the text.

# Word Distribution from Tom Sawyer

Word Frequency	Frequency of Frequency
1	3993
2	1292
3	664
4	410
5	243
6	199
7	172
8	131
9	82
10	91
11-50	540
51-100	99
> 100	102

- $TTR = 0.11 \Rightarrow$  Words occur on average 9 times each.
- But words have a very uneven distribution.

# Word Distribution from Tom Sawyer

Word Frequency	Frequency of Frequency
1	3993
2	1292
3	664
4	410
5	243
6	199
7	172
8	131
9	82
10	91
11-50	540
51-100	99
> 100	102

- $TTR = 0.11 \Rightarrow$  Words occur on average 9 times each.
- But words have a very uneven distribution.

## *Most words are rare*

- 3993 (50%) word types appear only once
- They are called *hapax legomena* (Greek for 'read only once')

# Word Distribution from Tom Sawyer

Word Frequency	Frequency of Frequency
1	3993
2	1292
3	664
4	410
5	243
6	199
7	172
8	131
9	82
10	91
11-50	540
51-100	99
> 100	102

- $TTR = 0.11 \Rightarrow$  Words occur on average 9 times each.
- But words have a very uneven distribution.

## *Most words are rare*

- 3993 (50%) word types appear only once
- They are called *hapax legomena* (Greek for 'read only once')

## *But common words are very common*

- 100 words account for 51% of all tokens of all text

# Zipf's Law

- Count the frequency of each word type in a large corpus
- List the word types in decreasing order of their frequency

NPTEL

# Zipf's Law

- Count the frequency of each word type in a large corpus
- List the word types in decreasing order of their frequency

## Zipf's Law

A relationship between the frequency of a word ( $f$ ) and its position in the list (its rank  $r$ ).



# Zipf's Law

- Count the frequency of each word type in a large corpus
- List the word types in decreasing order of their frequency

## Zipf's Law

A relationship between the frequency of a word ( $f$ ) and its position in the list (its rank  $r$ ).

$$f \propto \frac{1}{r}$$

# Zipf's Law

- Count the frequency of each word type in a large corpus
- List the word types in decreasing order of their frequency

## Zipf's Law

A relationship between the frequency of a word ( $f$ ) and its position in the list (its rank  $r$ ).

$$f \propto \frac{1}{r}$$

or, there is a constant  $k$  such that

$$f \cdot r = k$$

# Zipf's Law

- Count the frequency of each word type in a large corpus
- List the word types in decreasing order of their frequency

## Zipf's Law

A relationship between the frequency of a word ( $f$ ) and its position in the list (its rank  $r$ ).

$$f \propto \frac{1}{r}$$

or, there is a constant  $k$  such that

$$f \cdot r = k$$

i.e. the 50th most common word should occur with 3 times the frequency of the 150th most common word.

Let

- $p_r$  denote the probability of word of rank  $r$
- $N$  denote the total number of word occurrences

# Zipf's Law

Let

- $p_r$  denote the probability of word of rank  $r$
- $N$  denote the total number of word occurrences

$$p_r = \frac{f}{N} = \frac{A}{r}$$

# Zipf's Law

Let

- $p_r$  denote the probability of word of rank  $r$
- $N$  denote the total number of word occurrences

$$p_r = \frac{f}{N} = \frac{A}{r}$$

The value of  $A$  is found closer to 0.1 for corpus

# Empirical Evaluation from Tom Sawyer

Word	Freq. ( $f$ )	Rank ( $r$ )	$f \cdot r$	Word	Freq. ( $f$ )	Rank ( $r$ )	$f \cdot r$
the	3332	1	3332	turned	51	200	10200
and	2972	2	5944	you'll	30	300	9000
a	1775	3	5235	name	21	400	8400
he	877	10	8770	comes	16	500	8000
but	410	20	8400	group	13	600	7800
be	294	30	8820	lead	11	700	7700
there	222	40	8880	friends	10	800	8000
one	172	50	8600	begin	9	900	8100
about	158	60	9480	family	8	1000	8000
more	138	70	9660	brushed	4	2000	8000
never	124	80	9920	sins	2	3000	6000
Oh	116	90	10440	Could	2	4000	8000
two	104	100	10400	Applausive	1	8000	8000

## *Correlation: Number of meanings and word frequency*

The number of meanings  $m$  of a word obeys the law:

$$m \propto \sqrt{f}$$



# Zipf's Other Laws

## *Correlation: Number of meanings and word frequency*

The number of meanings  $m$  of a word obeys the law:

$$m \propto \sqrt{f}$$

Given the First law

$$m \propto \frac{1}{\sqrt{r}}$$

# Zipf's Other Laws

## *Correlation: Number of meanings and word frequency*

The number of meanings  $m$  of a word obeys the law:

$$m \propto \sqrt{f}$$

Given the First law

$$m \propto \frac{1}{\sqrt{r}}$$

## *Empirical Support*

- Rank  $\approx 10000$ , average 2.1 meanings
- Rank  $\approx 5000$ , average 3 meanings
- Rank  $\approx 2000$ , average 4.6 meanings

## *Correlation: Word length and word frequency*

Word frequency is inversely proportional to their length.

# Impact of Zipf's Law

## *The Good part*

Stopwords account for a large fraction of text, thus eliminating them greatly reduces the number of tokens in a text.

# Impact of Zipf's Law

## *The Good part*

Stopwords account for a large fraction of text, thus eliminating them greatly reduces the number of tokens in a text.

## *The Bad part*

Most words are extremely rare and thus, gathering sufficient data for meaningful statistical analysis is difficult for most words.

How does the size of the overall vocabulary (number of unique words) grow with the size of the corpus?

NPTEL

How does the size of the overall vocabulary (number of unique words) grow with the size of the corpus?

## *Heaps' Law*

Let  $|V|$  be the size of vocabulary and  $N$  be the number of tokens.

How does the size of the overall vocabulary (number of unique words) grow with the size of the corpus?

## *Heaps' Law*

Let  $|V|$  be the size of vocabulary and  $N$  be the number of tokens.

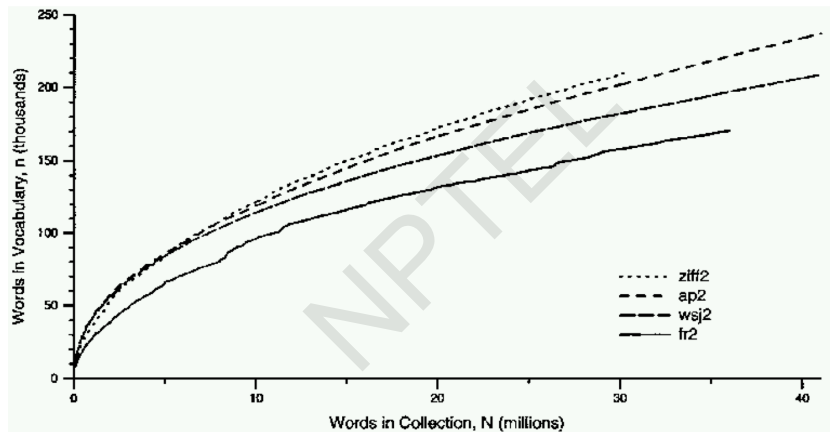
$$|V| = KN^\beta$$

Typically

- $K \approx 10\text{-}100$
- $\beta \approx 0.4 - 0.6$  (roughly square root)



# Heaps' Law: Empirical Evidence



# *Text Processing: Basics*

Pawan Goyal

CSE, IITKGP

Week 1: Lecture 5

## What is Tokenization?

Tokenization is the process of segmenting a string of characters into words.

Depending on the application in hand, you might have to perform *sentence segmentation* as well.

# Sentence Segmentation

The problem of deciding where the sentences begin and end.

## Challenges Involved

NPTEL

# Sentence Segmentation

The problem of deciding where the sentences begin and end.

## Challenges Involved

- While '!', '?' are quite unambiguous

# Sentence Segmentation

The problem of deciding where the sentences begin and end.

## Challenges Involved

- While '!', '?' are quite unambiguous
- Period "." is quite ambiguous and can be used additionally for
  - ▶ Abbreviations (Dr., Mr., m.p.h.)

# Sentence Segmentation

The problem of deciding where the sentences begin and end.

## Challenges Involved

- While '!', '?' are quite unambiguous
- Period "." is quite ambiguous and can be used additionally for
  - ▶ Abbreviations (Dr., Mr., m.p.h.)
  - ▶ Numbers (2.4%, 4.3)

# Sentence Segmentation

The problem of deciding where the sentences begin and end.

## Challenges Involved

- While '!', '?' are quite unambiguous
- Period "." is quite ambiguous and can be used additionally for
  - ▶ Abbreviations (Dr., Mr., m.p.h.)
  - ▶ Numbers (2.4%, 4.3)

## Approach: build a binary classifier

For each "."

- Decides EndOfSentence/NotEndOfSentence



# Sentence Segmentation

The problem of deciding where the sentences begin and end.

## Challenges Involved

- While '!', '?' are quite unambiguous
- Period "." is quite ambiguous and can be used additionally for
  - ▶ Abbreviations (Dr., Mr., m.p.h.)
  - ▶ Numbers (2.4%, 4.3)

## Approach: build a binary classifier

For each "."

- Decides EndOfSentence/NotEndOfSentence
- Classifiers can be: hand-written rules, regular expressions, or machine learning

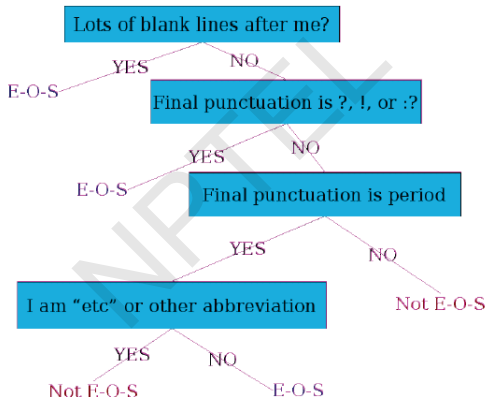
# *Sentence Segmentation: Decision Tree Example*

Decision Tree: Is this word the end-of-sentence (E-O-S)?

NPTEL

# Sentence Segmentation: Decision Tree Example

Decision Tree: Is this word the end-of-sentence (E-O-S)?



# *Other Important Features*

NPTEL

## Other Important Features

- Case of word with “.”: Upper, Lower, Cap, Number

# Other Important Features

- Case of word with “.”: Upper, Lower, Cap, Number
- Case of word after “.”: Upper, Lower, Cap, Number

# Other Important Features

- Case of word with “.”: Upper, Lower, Cap, Number
- Case of word after “.”: Upper, Lower, Cap, Number
- Numeric Features

# Other Important Features

- Case of word with “.”: Upper, Lower, Cap, Number
- Case of word after “.”: Upper, Lower, Cap, Number
- Numeric Features
  - ▶ Length of word with “.”



# Other Important Features

- Case of word with “.”: Upper, Lower, Cap, Number
- Case of word after “.”: Upper, Lower, Cap, Number
- Numeric Features
  - ▶ Length of word with “.”
  - ▶ Probability (word with “.” occurs at end-of-sentence)

# Other Important Features

- Case of word with “.”: Upper, Lower, Cap, Number
- Case of word after “.”: Upper, Lower, Cap, Number
- Numeric Features
  - ▶ Length of word with “.”
  - ▶ Probability (word with “.” occurs at end-of-sentence)
  - ▶ Probability (word after “.” occurs at beginning-of-sentence)

# Implementing Decision Trees

NPTEL

# Implementing Decision Trees

- Just an if-then-else statement

NPTEL

# Implementing Decision Trees

- Just an if-then-else statement
- Choosing the features is more important

NPTEL

# Implementing Decision Trees

- Just an if-then-else statement
- Choosing the features is more important
- For numeric features, thresholds are to be picked

# Implementing Decision Trees

- Just an if-then-else statement
- Choosing the features is more important
- For numeric features, thresholds are to be picked
- With increasing features including numerical ones, difficult to set up the structure by hand

# Implementing Decision Trees

- Just an if-then-else statement
- Choosing the features is more important
- For numeric features, thresholds are to be picked
- With increasing features including numerical ones, difficult to set up the structure by hand
- Decision Tree structure can be learned using machine learning over a training corpus



# Implementing Decision Trees

- Just an if-then-else statement
- Choosing the features is more important
- For numeric features, thresholds are to be picked
- With increasing features including numerical ones, difficult to set up the structure by hand
- Decision Tree structure can be learned using machine learning over a training corpus

## *Basic Idea*

Usually works top-down, by choosing a variable at each step that best splits the set of items.

Popular algorithms: ID3, C4.5, CART

# Other Classifiers

NPTEL

The questions in the decision tree can be thought of as features, that could be exploited by any other classifier:

The questions in the decision tree can be thought of as features, that could be exploited by any other classifier:

- Support Vector Machines
- Logistic regression
- Neural Networks

## *What is Tokenization?*

Tokenization is the process of segmenting a string of characters into words.

NPTEL

# Word Tokenization

## What is Tokenization?

Tokenization is the process of segmenting a string of characters into words.

*I have a can opener; but I can't open these cans.*

## Word Token

- An occurrence of a word
- For the above sentence, 11 word tokens.

## Word Type

- A different realization of a word
- For the above sentence, 10 word types.

# Tokenization in practice

- NLTK Toolkit (Python)
- Stanford CoreNLP (Java)
- Unix Commands

# Word Tokenization

NPTEL



## *Issues in Tokenization*

- Finland's → Finland Finlands Finland's ?
- What're, I'm, shouldn't → What are, I am, should not ?
- San Francisco → one token or two?
- m.p.h. → ??

## *Issues in Tokenization*

- Finland's → Finland Finlands Finland's ?
- What're, I'm, shouldn't → What are, I am, should not ?
- San Francisco → one token or two?
- m.p.h. → ??

For information retrieval, use the same convention for documents and queries

# Handling Hyphenation

Hyphens can be

NPTEL

# Handling Hyphenation

Hyphens can be

## *End-of-Line Hyphen*

Used for splitting whole words into part for text justification.

*This paper describes MIMIC, an adaptive mixed initia-tive spoken dialogue system that provides movie show-time information.*

# Handling Hyphenation

Hyphens can be

## *End-of-Line Hyphen*

Used for splitting whole words into part for text justification.

*This paper describes MIMIC, an adaptive mixed initiative spoken dialogue system that provides movie show-time information.*

## *Lexical Hyphen*

Certain prefixes are often written hyphenated, e.g. co-, pre-, meta-, multi-, etc.

# Handling Hyphenation

Hyphens can be

hyphen (-)

## End-of-Line Hyphen

Used for splitting whole words into part for text justification.

*This paper describes MIMIC, an adaptive mixed initiative spoken dialogue system that provides movie show-time information.*

## Lexical Hyphen

Certain prefixes are often written hyphenated, e.g. co-, pre-, meta-, multi-, etc.

## Sententially Determined Hyphenation

Mainly to prevent incorrect parsing of the phrase. Some possible usages:

- Noun modified by an 'ed'-verb: *case-based, hand-delivered*
- Entire expression as a modifier in a noun group: *three-to-five-year direct marketing plan*

# Language Specific Issues: French and German

## French

**l'ensemble:** want to match with un ensemble

# Language Specific Issues: French and German

## French

**l'ensemble:** want to match with un ensemble

## German

Noun compounds are not segmented

- Lebensversicherungsgesellschaftsangestellter
- 'life insurance company employee'
- Compound splitter required for German information retrieval



# Language Specific Issues: Chinese and Japanese

No space between words

莎拉波娃现在居住在美国东南部的佛罗里达。

莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达

Sharapova now lives in US southeastern Florida

# Language Specific Issues: Chinese and Japanese

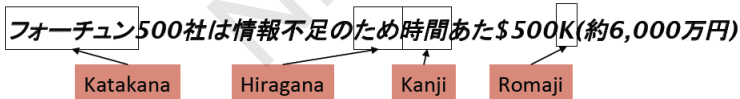
No space between words

莎拉波娃现在居住在美国东南部的佛罗里达。

莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达

Sharapova now lives in US southeastern Florida

**Japanese:** further complications with multiple alphabets intermingled.



सत्यम्ब्रूयात्प्रियम्ब्रूयान्नब्रूयात्सत्यमप्रियम्प्रियञ्चनानृतम्ब्रूयादेषधर्मःसनातनः

satyaṁbrūyātpriyaṁbrūyānnabrūyātsatyamapriyaṁpriyaṁcanānṛtambrūyād-  
eṣadharmahsanātanaḥ.

“One should tell the truth, one should say kind words; one should neither tell harsh truths, nor flattering lies; this is a rule for all times.”

सत्यम्ब्रूयात्प्रियम्ब्रूयान्ब्रूयात्सत्यमप्रियम्प्रियञ्चनानृतम्ब्रूयादेषधर्मःसनातनः

*satyaṁbrūyātpriyaṁbrūyānnabrūyātsatyamapriyaṁpriyaṁcanānṛtambrūyād-  
eṣadharmahsanātanaḥ.*

“One should tell the truth, one should say kind words; one should neither tell harsh truths, nor flattering lies; this is a rule for all times.”

## Segmented Text:

*satyam brūyāt priyam brūyāt na brūyāt satyam apriyam priyam ca na anṛtam  
brūyāt eṣaḥ dharmah sanātanaḥ.*

# Longest Words

Max ▾	Language (non scientific) ⇅
431	Sanskrit (Longest)
173	Greek
136	Afrikaans
85	Māori
79	German
74	Turkish
64	Icelandic
56	Hungarian
54	Spanish
49	Dutch
46	Malay
45	English

44	Romanian
42	Georgian
41	Czech
39	Bulgarian
39	Lithuanian
36	Kazakh
33	Norwegian
32	Tagalog
32	Polish
30	Serbian
30	Montenegrin
30	Italian
30	Croatian

Compound word composed of 431 letters, from the Varadāmbikā Parīṇaya Campū by Tirumalāmba

निरन्तरान्धकारिता-दिगन्तर-कन्दलदमन्द-सुधारस-बिन्दु-सान्द्रतर-घनाघन-वृन्द-सन्देहकर-  
स्यन्दमान-मकरन्द-बिन्दु-बन्धुरतर-माकन्द-तरु-कुल-तल्प-कल्प-मृदुल-सिकता-जाल-जटिल-  
मूल-तल-मरुवक-मिलदलघु-लघु-लय-कलित-रमणीय-पानीय-शालिका-बालिका-करार-विन्द-  
गलन्तिका-गलदेला-लवङ्ग-पाटल-घनसार-कस्तूरिकातिसौरभ-मेदुर-लघुतर-मधुर-शीतलतर-  
सलिलधारा-निराकरिष्णु-तदीय-विमल-विलोचन-मयूख-रेखापसारित-पिपासायास-पथिक-  
लोकान्

# Word Tokenization in Chinese or Sanskrit

Also called '**Word Segmentation**'.

NPTEL

# Word Tokenization in Chinese or Sanskrit

Also called '**Word Segmentation**'.

## *Greedy Algorithm for Chinese*

### **Maximum Matching (Greedy Algorithm)**

- Start a pointer at the beginning of the string
- Find the largest word in dictionary that matches the string starting at pointer
- Move the pointer over the word in string

*Think of the cases when word segmentation would be required for English Text.*



# Word Tokenization in Chinese or Sanskrit

Also called '**Word Segmentation**'.

## *Greedy Algorithm for Chinese*

### **Maximum Matching (Greedy Algorithm)**

- Start a pointer at the beginning of the string
- Find the largest word in dictionary that matches the string starting at pointer
- Move the pointer over the word in string

*Think of the cases when word segmentation would be required for English Text.*

Finding constituent words in a compound hashtags: #ThankYouSachin, #musicmonday etc.

# Text Segmentation for Sanskrit

## *General assumption behind the design*

Sentences from Classical Sanskrit may be generated by a regular relation  $R$  of the Kleene closure  $W^*$  of a regular set  $W$  of *words* over a finite alphabet  $\Sigma$ .

## General assumption behind the design

Sentences from Classical Sanskrit may be generated by a regular relation  $R$  of the Kleene closure  $W^*$  of a regular set  $W$  of words over a finite alphabet  $\Sigma$ .

- $W$ : vocabulary of (inflected) words (*padas*) and
- $R$ : sandhi

# Text Segmentation for Sanskrit

## General assumption behind the design

Sentences from Classical Sanskrit may be generated by a regular relation  $R$  of the Kleene closure  $W^*$  of a regular set  $W$  of words over a finite alphabet  $\Sigma$ .

- $W$ : vocabulary of (inflected) words (*padas*) and
- $R$ : sandhi

## Analysis of a sentence

A candidate sentence  $w$  is analyzed by inverting relation  $R$  to produce a finite sequence  $w_1, w_2, \dots, w_n$  of word forms, together with a proof that  $w \in R(w_1 \cdot w_2 \dots \cdot w_n)$ .

# Word Segmentation in Sanskrit

Sentence: सत्यम्ब्रूयात्प्रियम्ब्रूयान्नम्ब्रूयात्सत्यमप्रियम्प्रियञ्चनानृतम्ब्रूयादेषधर्मःसनातनः

✓Undo (120 Solutions)

satyambrūyātpriyambrūyānnabrūyātsatyam a priyamprīyañcanānṛtambrūyādeṣadharmasana ā tanah  
✓ ✓ ✓ ✓X ✓ ✓ ✓X ✓X ✓ ✓ ✓ ✓ ✓ ✓ ✓X  
brūyām satī ama ca na sana tanas  
✓X ✓X ✓X ✓X ✓X ✓X  
satī an san ata  
✓X ✓X ✓X ✓X ✓X ✓X  
sa na āta  
✓X ✓X ✓X ✓X ✓X ✓X  
a  
✓X ✓X

## Why to “normalize”?

Indexed text and query terms must have the same form.

- U.S.A. and USA should be matched

## Why to “normalize”?

Indexed text and query terms must have the same form.

- U.S.A. and USA should be matched
- We implicitly define equivalence classes of terms

- Reduce all letters to lower case

NPTEL



- Reduce all letters to lower case
- Possible exceptions (Task dependent):
  - ▶ Upper case in mid sentence, may point to named entities (e.g. General Motors)

- Reduce all letters to lower case
- Possible exceptions (Task dependent):
  - ▶ Upper case in mid sentence, may point to named entities (e.g. General Motors)
  - ▶ For MT and information extraction, some cases might be helpful (*US* vs. *us*)

# Lemmatization

reduces words to their base or dictionary form, also known as the lemma

- Reduce inflections or variant forms to base form:
  - ▶ am, are, is → be
  - ▶ car, cars, car's, cars' → car
- Have to find the correct dictionary headword form

Morphology studies the internal structure of words, how words are built up from smaller meaningful units called **morphemes**

NPTEL

Morphology studies the internal structure of words, how words are built up from smaller meaningful units called **morphemes**

*Morphemes are divided into two categories*

- Stems: The core meaning bearing units
- Affixes: Bits and pieces adhering to stems to change their meanings and grammatical functions

Morphology studies the internal structure of words, how words are built up from smaller meaningful units called **morphemes**

*Morphemes are divided into two categories*

- Stems: The core meaning bearing units
- Affixes: Bits and pieces adhering to stems to change their meanings and grammatical functions
  - ▶ Prefix: un-, anti-, etc (a-, ati-, pra- etc.)

Morphology studies the internal structure of words, how words are built up from smaller meaningful units called **morphemes**

*Morphemes are divided into two categories*

- Stems: The core meaning bearing units
- Affixes: Bits and pieces adhering to stems to change their meanings and grammatical functions
  - ▶ Prefix: un-, anti-, etc (a-, ati-, pra- etc.)
  - ▶ Suffix: -ity, -ation, etc (-taa, -ke, -ka etc.)

Morphology studies the internal structure of words, how words are built up from smaller meaningful units called **morphemes**

*Morphemes are divided into two categories*

- Stems: The core meaning bearing units
- Affixes: Bits and pieces adhering to stems to change their meanings and grammatical functions
  - ▶ Prefix: un-, anti-, etc (a-, ati-, pra- etc.)
  - ▶ Suffix: -ity, -ation, etc (-taa, -ke, -ka etc.)
  - ▶ Infix: 'n' in 'vindati' (he knows), as contrasted with *vid* (to know).



- Reducing terms to their stems, used in information retrieval

NPTEL

- Reducing terms to their stems, used in information retrieval
- Crude chopping of affixes
  - ▶ language dependent

NPTEL

- Reducing terms to their stems, used in information retrieval
- Crude chopping of affixes
  - ▶ language dependent
  - ▶ *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 equal to compress

# Porter's algorithm

## Step 1a

- sses  $\rightarrow$  ss (caresses  $\rightarrow$  caress)
- ies  $\rightarrow$  i (ponies  $\rightarrow$  poni)
- ss  $\rightarrow$  ss (caress  $\rightarrow$  caress)
- s  $\rightarrow$   $\phi$  (cats  $\rightarrow$  cat)

# Porter's algorithm

## Step 1a

- sses  $\rightarrow$  ss (caresses  $\rightarrow$  caress)
- ies  $\rightarrow$  i (ponies  $\rightarrow$  poni)
- ss  $\rightarrow$  ss (caress  $\rightarrow$  caress)
- s  $\rightarrow$   $\phi$  (cats  $\rightarrow$  cat)

## Step 1b

- (\*v\*)ing  $\rightarrow$   $\phi$  (walking  $\rightarrow$  walk, king  $\rightarrow$

# Porter's algorithm

## Step 1a

- sses  $\rightarrow$  ss (caresses  $\rightarrow$  caress)
- ies  $\rightarrow$  i (ponies  $\rightarrow$  poni)
- ss  $\rightarrow$  ss (caress  $\rightarrow$  caress)
- s  $\rightarrow$   $\phi$  (cats  $\rightarrow$  cat)

## Step 1b

- (\*v\*)ing  $\rightarrow$   $\phi$  (walking  $\rightarrow$  walk, king  $\rightarrow$  king)
- (\*v\*)ed  $\rightarrow$   $\phi$  (played  $\rightarrow$  play)

# Porter's algorithm

## Step 2

- ational → ate (relational → relate)
- izer → ize (digitizer → digitize)
- ator → ate (operator → operate)

# Porter's algorithm

## Step 2

- ational  $\rightarrow$  ate (relational  $\rightarrow$  relate)
- izer  $\rightarrow$  ize (digitizer  $\rightarrow$  digitize)
- ator  $\rightarrow$  ate (operator  $\rightarrow$  operate)

## Step 3

- al  $\rightarrow$   $\phi$  (revival  $\rightarrow$  reviv)
- able  $\rightarrow$   $\phi$  (adjustable  $\rightarrow$  adjust)
- ate  $\rightarrow$   $\phi$  (activate  $\rightarrow$  activ)