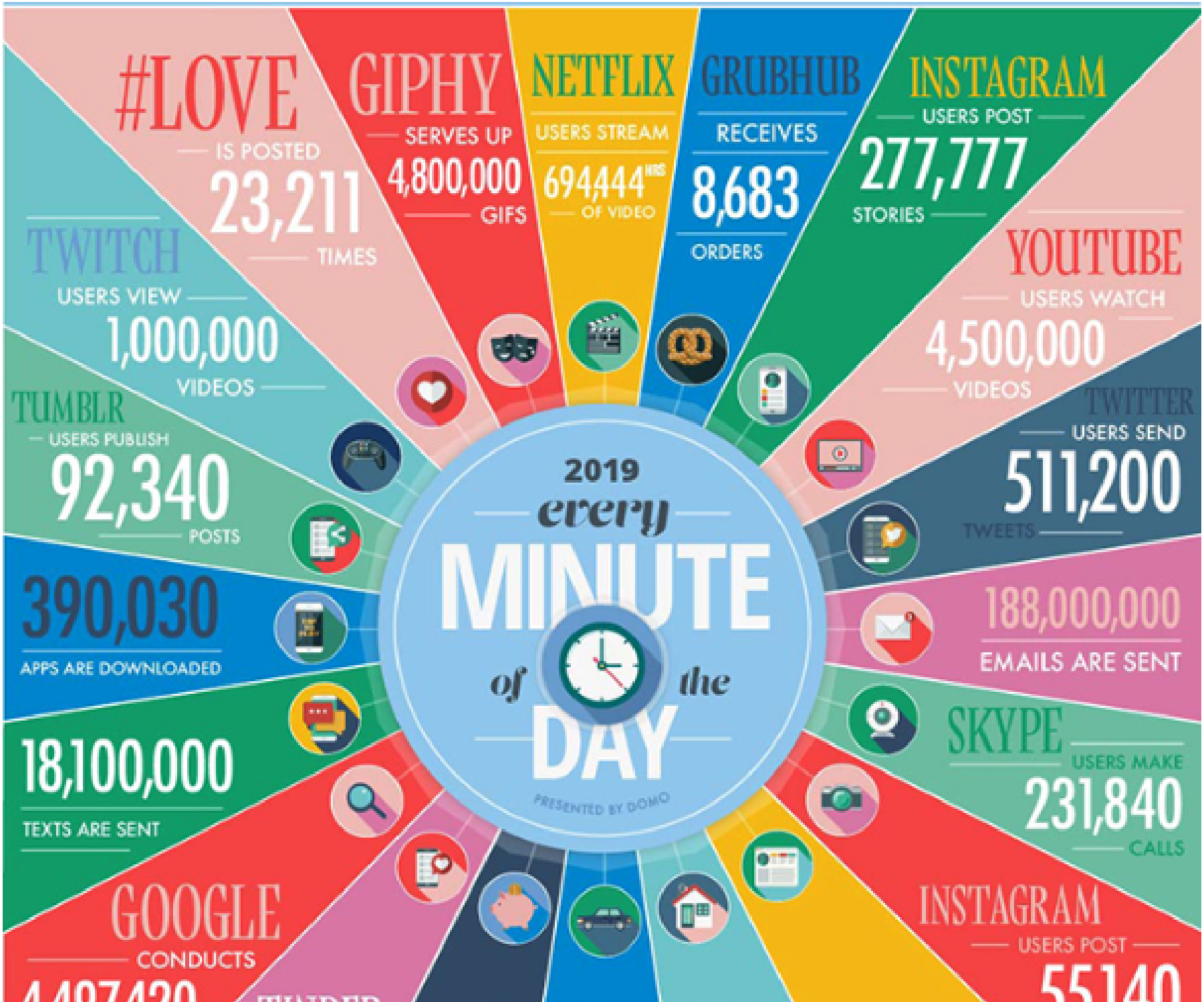
Welcome to 2nd year project:
Natural Language Processing and Deep Learning

Who are we?

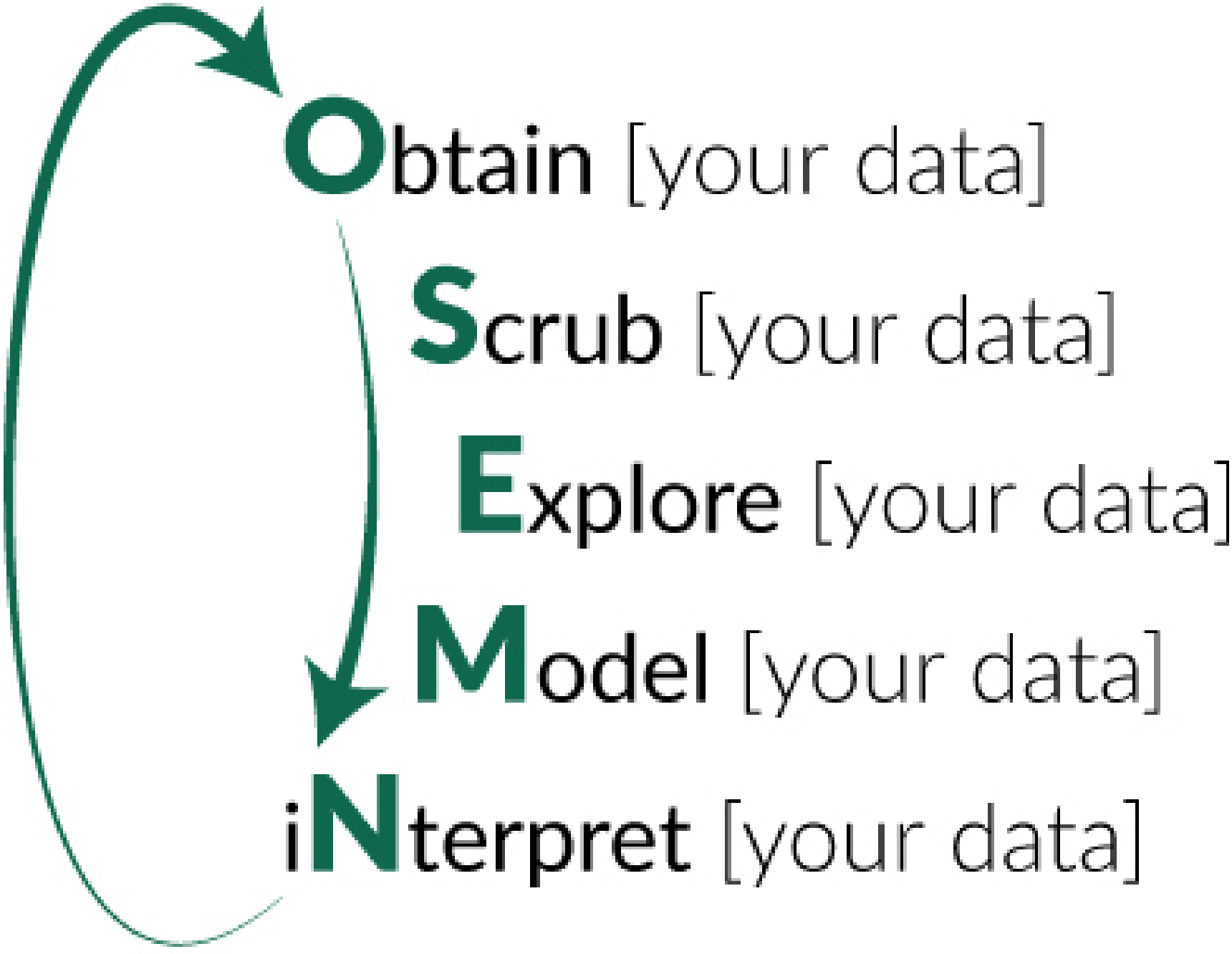
Data Never Sleeps

E.g., In 2019, how many tweets were send every minute?
Go to www.menti.com and use the code 451093


#LOVE
IS POSTED
23,211
TIMES
GIPHY
SERVES UP
4,800,000
GIFS
NETFLIX
USERS STREAM
694,444
OF VIDEO
GRUBHUB
RECEIVES
8,683
ORDERS
INSTAGRAM
USERS POST
277,777
STORIES
TWITCH
USERS VIEW
1,000,000
VIDEOS
YOUTUBE
USERS WATCH
4,500,000
VIDEOS
TUMBLR
USERS PUBLISH
92,340
POSTS
TWITTER
USERS SEND
511,200
TWEETS
2019
every
MINUTE
of the
DAY
PRESENTED BY DOMO
390,030
APPS ARE DOWNLOADED
188,000,000
EMAILS ARE SENT
18,100,000
TEXTS ARE SENT
SKYPE
USERS MAKE
231,840
CALLS
GOOGLE
CONDUCTS
INSTAGRAM
USERS POST
55140

Data Science is OSEMN!

- OSEMN model (Hilary Mason, 2010)
- Pronounced 'awesome'


Obtain [your data]
Scrub [your data]
Explore [your data]
Model [your data]
iNterpret [your data]

What is Natural Language Processing (NLP)?

- An interdisciplinary research field Goal: enabling computers to understand and generate natural language, just as we humans do
- Deep understanding of broad language
  - not just string processing or keyword matching
- A lot of data out there is unstructured, free text

NLP is not just fascinating, it is important:

- Improve communication
- Extract information
- More and more digital data available: NLP is necessary to analyze larger amounts
- Not just linguistics insights, also digital humanities

NLP is not just fascinating, it is important:

- Improve communication
- Extract information
- More and more digital data available: NLP is necessary to analyze larger amounts
- Not just linguistics insights, also digital humanities
- Avoid injuries

Course practicalities

Teachers

- Rob van der Goot: `https://robvanderg.github.io/`
- Christian Hardmeier: `https://christianhardmeier.rax.ch/`
- Max Müller-Eberstein: `https://personads.me/`
- Elisa Bassignana: `https://elisabassignana.github.io/`
- Dirk Hovy: `https://dirkhovy.com/` (Guest lecturer)

Teaching assistants

- Cathrine Damgaard
- Trine Naja Eriksen
- Ludek Cizinský

Prerequisites:

We are assuming you also take the Large-Scale Data Analysis course and:

- Are a bit familiar with command line Linux (or try `https://missing.csail.mit.edu/2020/course-shell/` and `https://missing.csail.mit.edu/2020/shell-tools/`)
- Know how to use the HPC

Course outcomes

- Discuss, clearly explain, and reflect upon central concepts, algorithms, and challenges in natural language processing (NLP) and deep learning (DL).
- Organize, plan, and carry out collaborative work in a smaller project group.
- Obtain, scrub, explore and preprocess a wide range of relevant raw data for a given problem. Identify and analyze the relevant options for data collection and preprocessing and select the most suitable ones.
- Design and implement a sound experiment in NLP.
- Distinguish and evaluate the advantages of different design choices or approaches to the same task (e.g., traditional versus deep-learning based solutions).
- Evaluate the achieved solution and carry out a detailed error analysis, relating the findings back to the overall problem domain
- Explain in writing (project group report) adhering to academic standards in writing.

Part I (week 5-11): Lecture phase

- Lectures:
- Monday 14:00-16:00
- Wednesday 08:00-10:00
- Labs:
- Monday 16:00-18:00
- Thursday 08:00-10:00

Course Materials

- Main textbook: Jurafsky and Martin (3rd edition 2021): `https://web.stanford.edu/~jurafsky/slp3/old_dec21/indexdec21.html`
- Note that this is not the latest version (as it is still being updated)
- Main reading material on LearnIt, will be the included in the exam syllabus. Background reading is also on LearnIt, and more might be given in the slides.
- We use github for the slides and assignments: `https://github.itu.dk/robv/intro-nlp2023`
- We also have a Slack channel, to which you already should have been invited (otherwise find link on LearnIt)

Lecture Preparation

- Read the provided reading material
- Do the exercises in the lab!
- There might be quizzes in class

Course Assessment and Course structure

- Part I (lectures and labs):
- prepares you for the project phase
- Solve $>= 5/6$ assignments $\mapsto$ $>=10$
- Labs include exercises to help get you deeper into the material
- Hand in before before first lecture next week (Monday 14:00)
- Homework feedback will be during the first lab (Monday 16:00)

Group formation (week 11): Project phase starts

- Group formation day and project kick-off day on March 15 (reserve that day) Part II (week 12-21): Project phase
- No lectures, but labs are available for feedback on the project.
- Group presentations project proposals (29-03 and 03-04)

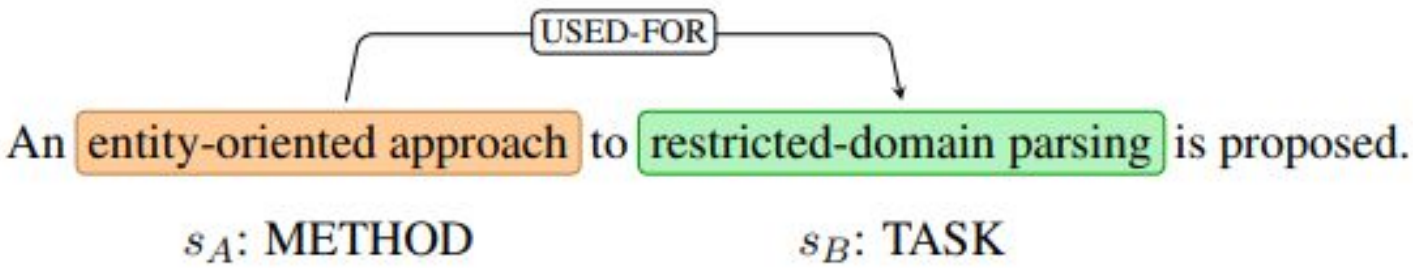
Course Project: Named Entity Recognition or Relation Extraction

- Identifying entities in text is an important step to interpretation of language and extraction of information. We will take a closer look at NER and RE in the lectures, and identify open problems which you can tackle in your project.

Named Entity Recognition (NER)

[Johann Adam Birkenstock]$_{PER}$ in 1774 founded [Birkenstock]$_{ORG}$ shoe company

Relation Extraction



Course Project: Named Entity Recognition or Relation Extraction

Methods involved, amongst others:

- Data scrubbing and cleaning (processing)
- Modeling: Comparison of traditional ML versus Neural Approaches, Error Analysis
- NLP in the wild
- Eploiting additional (raw) data
- Can humans perform these tasks?

Assessment

- Final project, to be completed in a group of 4 students
- Hand-in: May 26, 2022 at 14:00. this is a preset, fixed date, no extensions!

Final project

- Topics will be suggested, you can also pick your own (within NER and RE).
- Max 5 pages ACL style files and code.
- You will answer a new research question.
- In many ways similar to the papers we read/discuss during the course.
- We will have a lecture on how to write an great NLP paper (according to Rob).
- Oppurtunity for additional feedback: hand in a week earlier

Exam

- Group presentation (slides) based on group report
- Individual oral exam: small part on project and a random topic from the exam syllabus
- External examiner
- Exam dates:
  - June 20-23, 2022 – reserve these dates in your calendar
- Re-exam: individual project with oral exam

Late Hand-In

- Late hand-ins cannot be accepted
- Exceptions can be made in rare cases, but only through SAP
- Get in touch with SAP at least one working day in advance
  see: `https://studyguide.itu.dk/ds/your-programme/exams/illness`

Plagiarism

- Don't do it
- Don't enable it
- Check: `https://itustudent.itu.dk/study-administration/exams/academic-misconduct/`

Semester Schedule

Spring 2023	4th semester DS				
Week#	Lecture#		LASDA (Maria) Fri	DVDDM (Michele) Thu	SEYEP (Rob) Mon/Wed
5	1	Feb			
6	2				
7	3				
8	4				
9	5	Mar			
10	6		deadline assignment 1		
11	7				
12	8				Start project phase
13	9		deadline assignment 2		
14	10	Apr	Easter	Easter	Easter
15	10				Easter
16	11				project proposal + presentation
17	12				
18	13	May	Prayer Day		
19	14		deadline assignment 3		
20				Exam Submission 19/5	
21			exam preparation		Exam Submission 26/5
22		June	Exam Submission 2/6		
23				Oral exam 7-14/6	
24				Oral exam 7-14/6	
25					Oral exam 20-23/6
26					

Python

- Lectures, lab exercises and assignments focus on Python3
- Python is a leading language for data science, machine learning etc., with many relevant libraries
- Labs and assignments focus on the development of a mix of standalone Python code, and use of Unix command line tools.

Important Dates - Summary

Note: keep an eye on LearnIt for details/updates

- Beginning of February to medio of March - Lecture Phase
- 15-03: group formation day & Project kick-off
- 29-03 and 03-04: project proposal presentations
- 27-03: upload baseline predictions
- 12-04: project proposal deadline
- 11-05: optionally upload draft
- 19-05: final project upload

How to reach us?

Slack

- We have a dedicated Slack channel: `https://2ndyearproject-2023.slack.com/` please post questions there (instead of private emails)

1. Introduction to NLP: Tokenization and Regular Expressions
2. Command Line Processing of Text, Experimental Standards
3. Dataset Annotation and POS Tagging
4. Machine Learning Classification in NLP
5. N-gram Language Models and Traditional Sequence Labeling
6. Introduction to Neural Networks for Text and Words as Embeddings
7. word2vec
8. Convolutional Neural Networks and Relation Extraction
9. Neural Language Models and RNN 1
10. RNN 2 (LSTM, bi-, GRU)
11. Introduction to Contextualized Language Models (ELMO)
12. Contextual Language Models 2 (BERT)
13. Bias in NLP

Any questions?

Introduction to Natural Language Processing (NLP)

After this lecture you should:

- know what Natural Language Processing (NLP) is and why language is so challenging
- have refreshed your memory on regular expressions
- be able to implement a tokenizer and identify issues for tokenization

What's so special about human language?

- most important *distinctive* human characteristic
- the hard part in AI (intelligence)
- communication is/was central in human development

NLP: Where are we now?

- Can you think of an NLP application that you use regularly?*

Do you know these...?

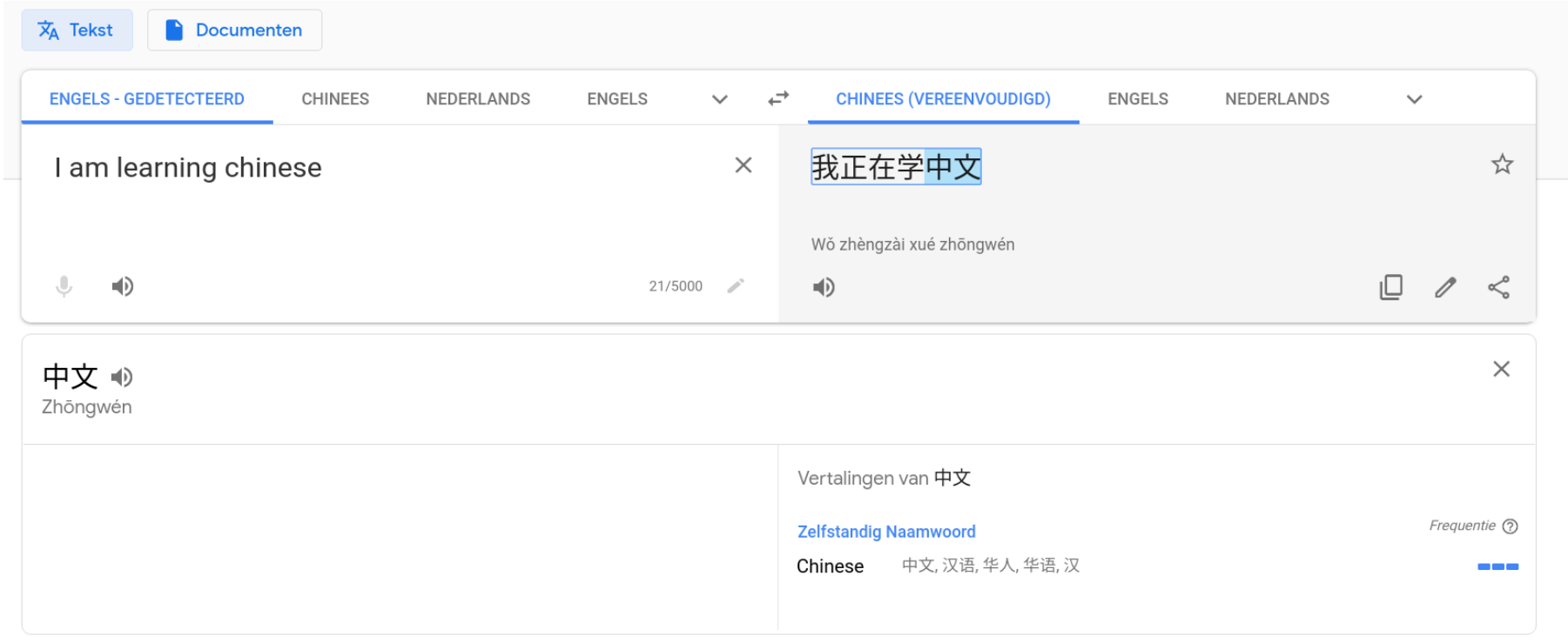
Personal assistants - Everywhere!




Speech Recognition

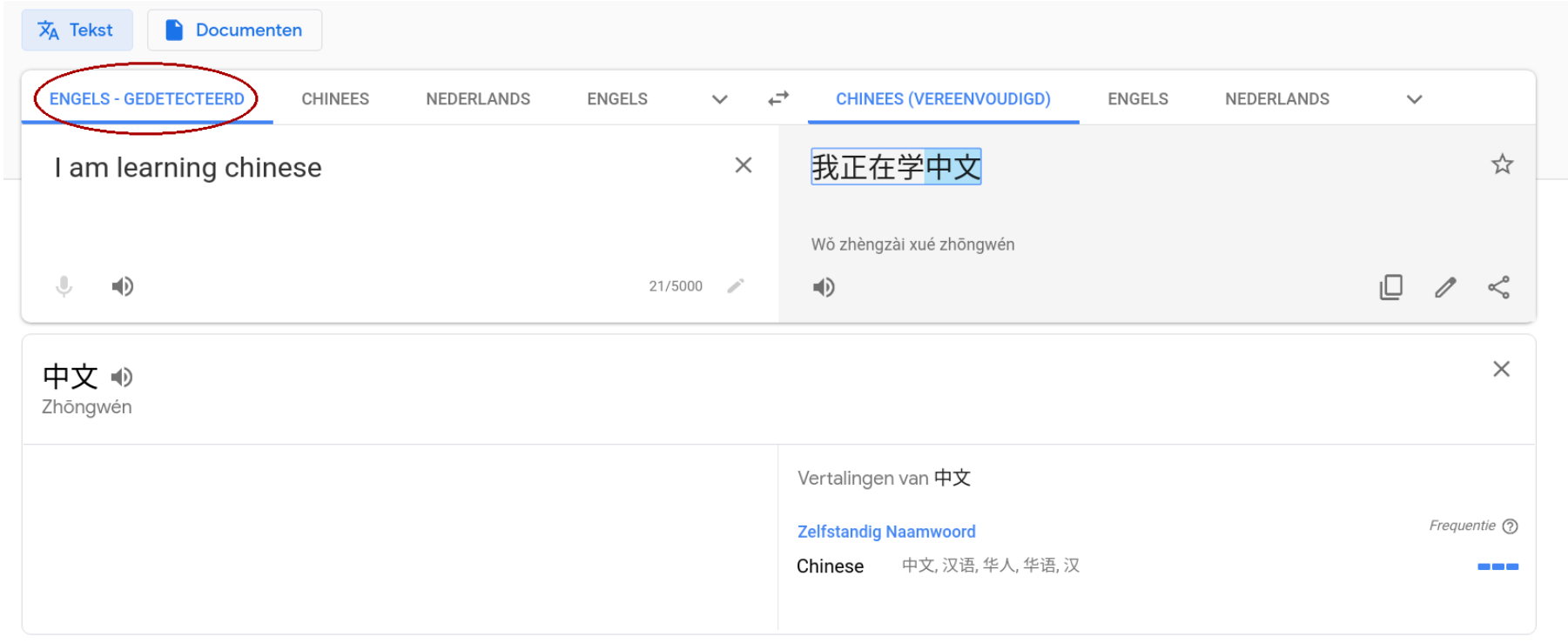
Speech Recognition is usually not considered core NLP.

Machine Translation



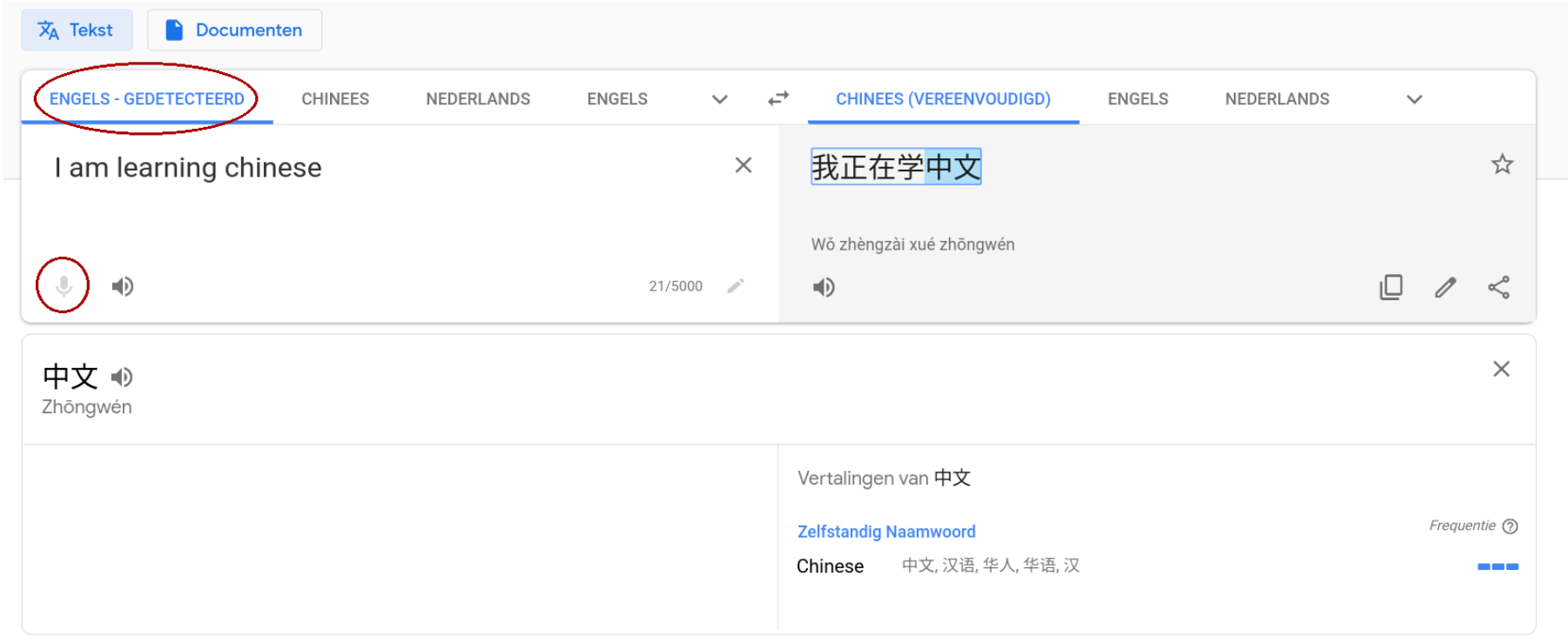
http://translate.google.com/

Machine Translation



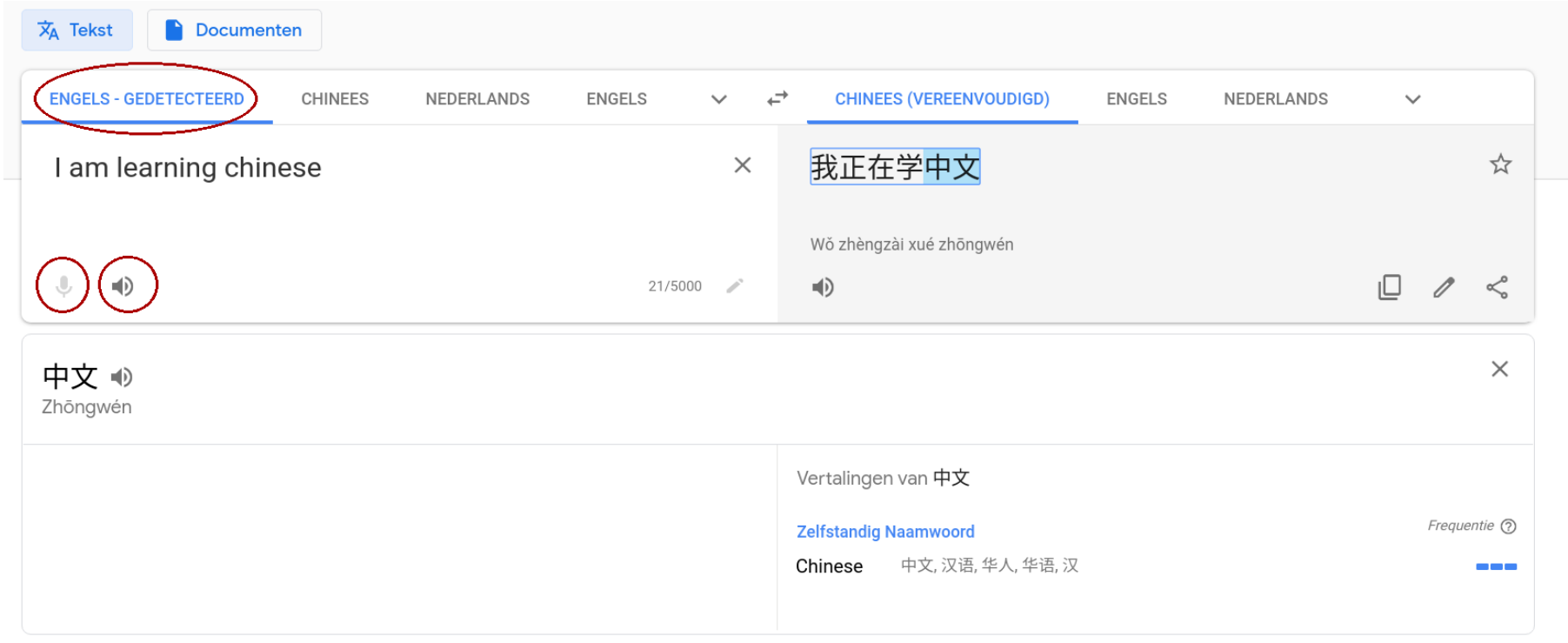
http://translate.google.com/

Machine Translation



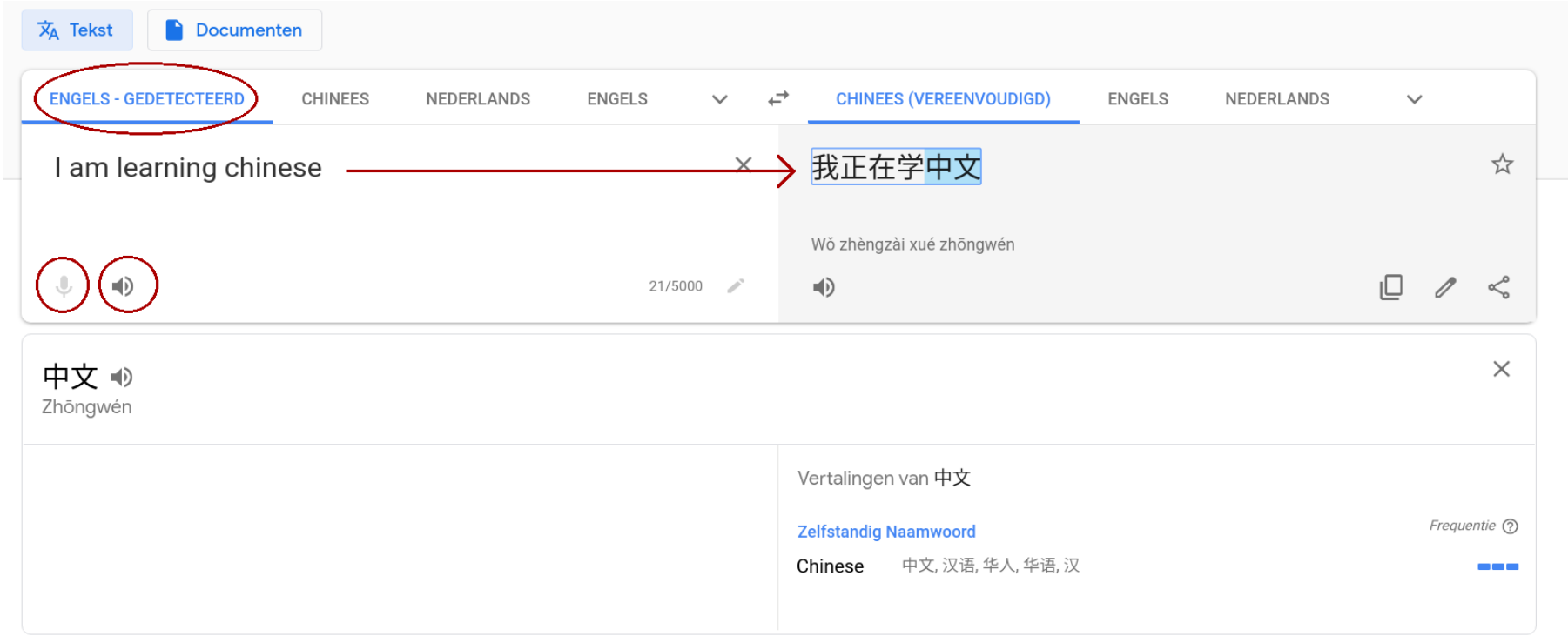
http://translate.google.com/

Machine Translation



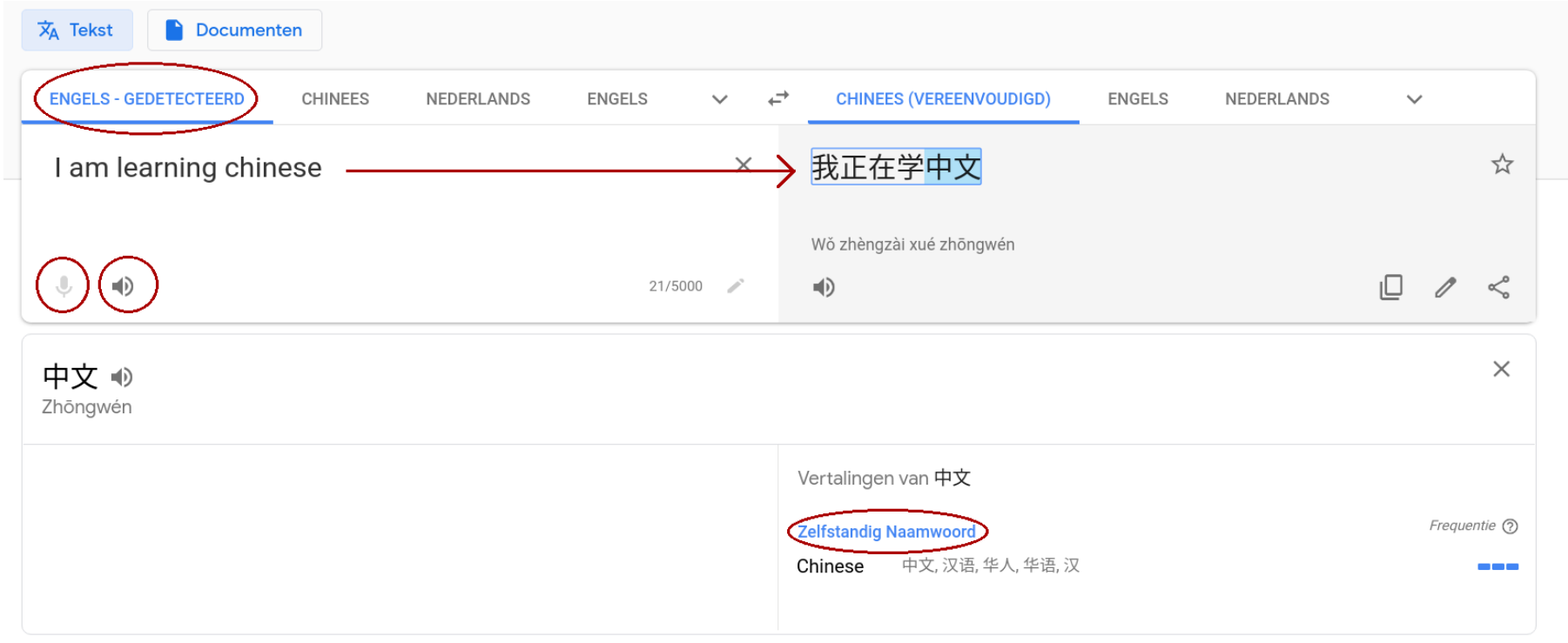
http://translate.google.com/

Machine Translation



http://translate.google.com/

Machine Translation



http://translate.google.com/

Information Extraction

andrew mccallum

Web Images Maps Shopping More Search tools

About 4,380,000 results (0.20 seconds)

Cookies help us deliver our services. By using our services, you agree to our use of cookies.

OK Learn more

Andrew McCallum Homepage
www.cs.umass.edu/~**mccallum**
Machine learning, text and information retrieval and extraction, reinforcement learning.
Andrew McCallum Publications - Andrew McCallum Bio - People - Teaching

Andrew McCallum - London Metropolitan University
www.londonmet.ac.uk/faculties/faculty-of...k.../**andrew-mccallum**/
Andrew taught English in London secondary schools for 15 years before coming to London Met in 2008. He is course tutor for the PGCE in Secondary English ...

Andrew McCallum - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Andrew_McCallum**
Andrew McCallum is a professor and researcher in the computer science department at University of Massachusetts Amherst. His primary specialties are in ...

Andrew Mccallum - United Kingdom profiles | LinkedIn
uk.linkedin.com/pub/dir/**Andrew**/**Mccallum**
View the profiles of professionals on LinkedIn named **Andrew Mccallum** located in the United Kingdom. There are 25 professionals named **Andrew Mccallum** in ...

Andrew McCallum

Software Developer

Andrew McCallum is a professor and researcher in the computer science department at University of Massachusetts Amherst. Wikipedia

Education: Dartmouth College, University of Rochester

Awards: Best 10-year Paper Award of the ICML

People also search for


Tom M. Mitchell


Lee Giles

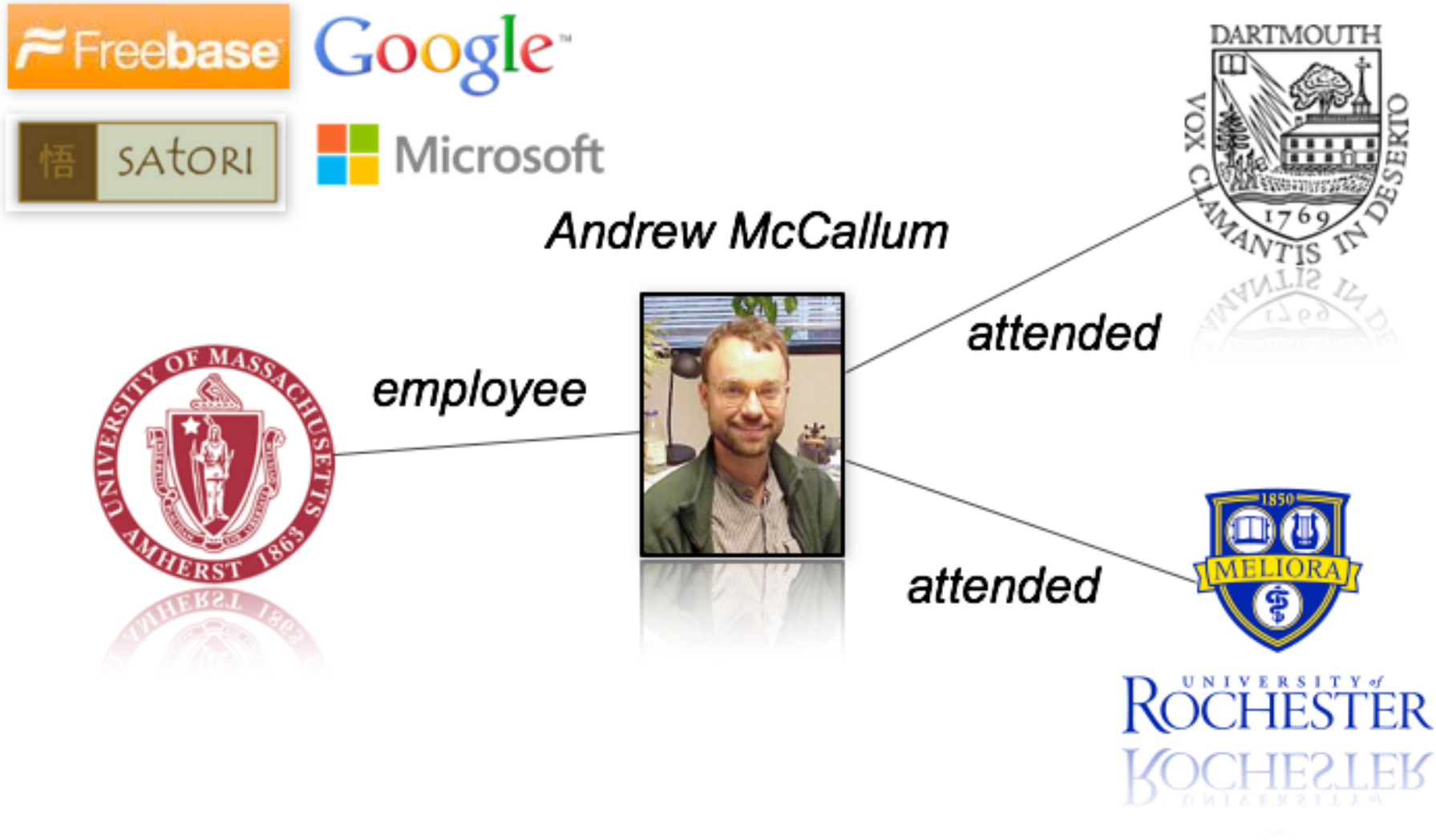

David M. Blei


Michael Collins

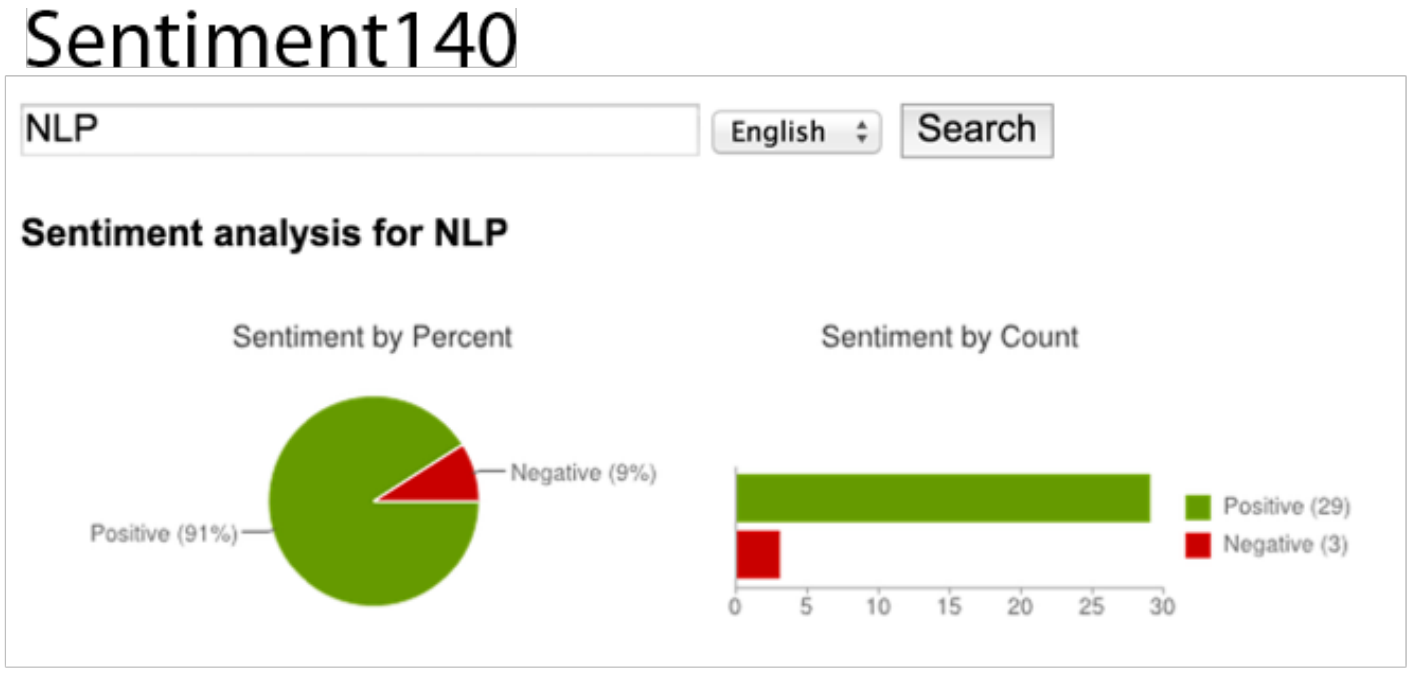

Robert Schapire

Feedback/More info

Information Extraction



Sentiment Analysis



JordanBone1: Recently enrolled on to a **NLP** life coaching course & now an online wedding planning course! ???? I love groupon & wowcher lol
Posted: 6 hours ago

BluffMasterPUA: this guy on the bus next to me was breaking some chick down with logic. Crazy **NLP** but he looked like a chode.
Posted: 9 hours ago

Why is it so difficult?

Are these sentences ok? what do they mean?

Why is it so difficult?

Are these sentences ok? what do they mean?
It's not a bird, it's not a plane,
It must be Dave who's on the train

Why is it so difficult?

Are these sentences ok? what do they mean?
It's not a bird, it's not a plane,
It must be Dave who's on the train

3 AM!
The painted cow !
Yeaaaaaaaaaaah!!
You ain't stoppin' us now !
Yeah ! I am the Junglist souljah.
Come On!
The rocket launcher stopped ya! (HEY!)
It's not a bird, it's not a plane,
It must be Dave who's on the train
Wanna wanna get'cha, gonna gonna get'cha!
Tell 'em that I told ya
Yeah!

What about these sentences?

- The cat sat on the mat.
- The mat sat on the cat.

What about these sentences?

- The cat sat on the mat.
- The mat sat on the cat.
- Sat the cat mat the on.

What about these sentences?

- The cat sat on the mat.
- The mat sat on the cat.
- Sat the cat mat the on.
- Mary went store.
- Mary went storing.

What about these sentences?

- The cat sat on the mat.
- The mat sat on the cat.
- Sat the cat mat the on.
- Mary went store.
- Mary went storing.
- The store went to Mary.

What is so difficult about NLP?

Example: *I made her duck*
What is the meaning of this sentence? (Is there only one meaning?)

1. I cooked a duck for her
2. I cooked a duck that belonged to her
3. I created a (wooden?) duck she owns
4. I caused her to quickly lower her upper body
5. I turned her into a duck (magic!)

Ambiguity is everywhere

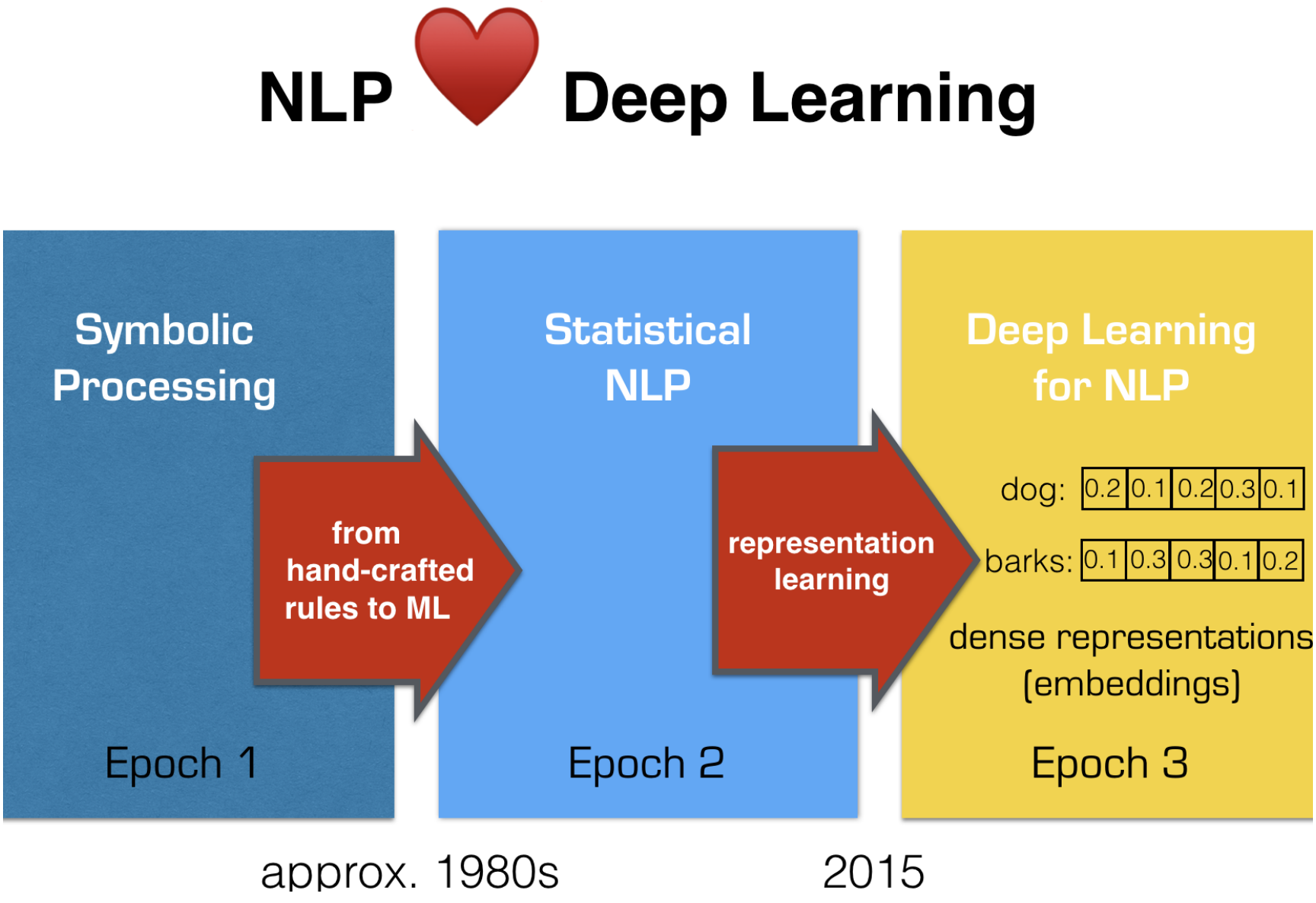
- Nancy Pelosi Calls for -Trump to be Removed from Office-
- -Nancy Pelosi- Calls for Trump -to be Removed from Office-

Ambiguity of language is manifested at many linguistic levels:

- lexical
- syntax
- semantics
- pragmatics

Further challenges: multilinguality, morphology ...

Barbara Planks one slide history of the field:



Types of tasks in NLP (non-exhaustive)

- text classification: sentiment, topic, intent, stance, language(variety)
- relation between texts: text similarity, textual entailment
- relation between words: syntactic parsing, relation extraction, coreference resolution
- sequence labeling: Parts-Of-Speech, named entity recognition, semantic role labeling, language(variety)
- text generation: machine translation, question answering, dialogue, data to text, summarization*
- transformations: grammatical error correction, tokenization, lemmatization

Types of tasks in NLP (non-exhaustive)

- text classification: sentiment, topic, intent, stance, language(variety)
- relation between texts: text similarity, textual entailment
- relation between words: syntactic parsing, relation extraction, coreference resolution
- sequence labeling: Parts-Of-Speech, named entity recognition, semantic role labeling, language(variety)
- text generation: machine translation, question answering, dialogue, data to text, summarization*
- transformations: grammatical error correction, tokenization, lemmatization

Beyond:

- Speech recognition
- Image recognition
- Information retrieval

Types of tasks in NLP

Other division:

- syntactic: usually considered a first step to interpretation
- semantic: deal with meaning

Input	I	saw	the	light.	
text classification					English
relation between texts					I saw the darkness. $\mapsto$ contradict
relation between words	2,nsubj	0,root	4,det	2,obj	
sequence labeling	pron	verb	det	noun	
generation	jeg så lyset.				
transformations	I	see	the	light	.

Input	I	saw	the	light.	
text classification					English
relation between texts					I saw the darkness. $\mapsto$ contradict
relation between words	2,nsubj	0,root	4,det	2,obj	
sequence labeling	pron	verb	det	noun	
generation	jeg så lyset.				
transformations	I	see	the	light	.

Go to www.menti.com and use the code 451093

What's a word? - Tokenization

- tokenization is to identify the words in a string of characters.

In Python you can tokenize a text via ‘split()‘:

In Python you can tokenize a text via ‘split()‘:

```
text = """Mr. Bob Dobolina is thinkin’ of a master plan.
Why doesn’t he quit?"""
text.split(" ")
```

OUTPUT:

```
['Mr.',
 'Bob',
 'Dobolina',
 'is',
 "thinkin'",
 'of',
 'a',
 'master',
 'plan.\nWhy',
 "doesn't",
 'he',
 'quit?']
```

OUTPUT:

```
['Mr.',
 'Bob',
 'Dobolina',
 'is',
 "thinkin'",
 'of',
 'a',
 'master',
 'plan.\nWhy',
 "doesn't",
 'he',
 'quit?']
```

Why is this suboptimal?

Python allows users to construct tokenizers using Regular Expressions

- A formal language for specifying text strings (an algebraic notation for characterizing a set of strings)
- Can be used to find subsets of text, or in tokenization to define patterns at which to split tokens.

Regular Expressions: Disjunctions

- Letters inside square brackets []

Pattern	Matches
`[wW]oodchuck`	Woodchuck, woodchuck
`[1234567890]`	Any digit

- Ranges `[A-Z]`

Pattern	Matches	
`[A-Z]`	An upper case letter	`Drenched Blossoms`
`[a-z]`	A lower case letter	`my beans were impatient`
`[0-9]`	A single digit	`Chapter 1: Down the Rabbit Hole`

Slides from Speech and Language Processing

Regular Expressions: Negation in Disjunction

- Negations `[^Ss]`
  - Carat means negation only when first in []

Pattern	Matches	
`[^A-Z]`	Not an upper case letter	`Oyfn pripetchik`
`[^Ss]`	Neither 'S' nor 's'	`I have no exquisite reason"`
`a^b`	The pattern a carat b	`Look up a^b now`

Slides from Speech and Language Processing

Regular Expressions: More Disjunction

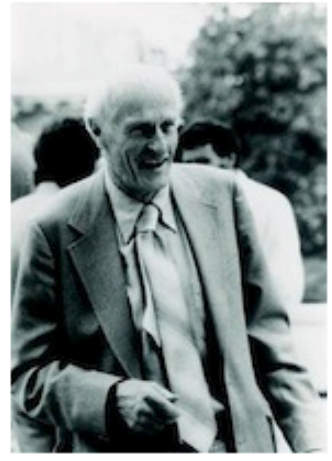
- Woodchucks is another name for groundhog!
- The pipe | for disjunction

Pattern	Matches
`groundhog|woodchuck`	
`yours|mine`	yours mine
`a|b|c`	= [abc]
`[gG]roundhog|[Ww]oodchuck`	

Slides from Speech and Language Processing

Regular Expressions: ? * + .

Pattern	Matches	
colou?r	Optional previous char	color colour
oo*h!	0 or more of previous char	oh! ooh! oooh! ooooh!
o+h!	1 or more of previous char	oh! ooh! oooh! ooooh!
baa+		baa baaa baaaa baaaaa
beg.n		begin begun begun beg3n



Stephen C Kleene

Kleene *, Kleene +

Slides from Speech and Language Processing

Regular Expressions: Anchors ^ $

Pattern	Matches
^[A-Z]	Palo Alto
^[^A-Za-z]	1 end "Hello"
\.$	The end.
.$	The end? The end!

Slides from Speech and Language Processing

Example

- Find me all instances of the word "the" in a text.

```
the
```

Misses capitalized examples

```
[tT]he
```

Incorrectly returns `other` or `theology`

```
[^a-zA-Z][tT]he[^a-zA-Z]
```

Slides from Speech and Language Processing

A regular expression is a compact definition of a set of (character) sequences.Examples:

- "Mr": set containing only "Mr."
- "[abc]": set containing only the characters 'a', 'b' and 'c'
- "\s": set of all whitespace characters
- "1+": set of all sequences of at least one '1'
- etc.

```
import re
re.compile('\s').split(text)
```

OUTPUT:

```
['Mr.',
 'Bob',
 'Dobolina',
 'is',
 "thinkin'",
 'of',
 'a',
 'master',
 'plan.',
 'Why',
 "doesn't",
 'he',
 'quit?']
```

Problems:

- Bad treatment of punctuation.
- It might be easier to define a token than a gap.

Problems:

- Bad treatment of punctuation.
- It might be easier to define a token than a gap.

Let us use ‘findall‘ instead:

```
re.compile('\w+|[\.?]').findall(text)
```

OUTPUT:

```
['Mr',
 '.',
 'Bob',
 'Dobolina',
 'is',
 'thinkin',
 'of',
 'a',
 'master',
 'plan',
 '.',
 'Why',
 'doesn',
 't',
 'he',
 'quit',
 '?']
```

Problems:

- "Mr." is split into two tokens, should be single.
- Lost an apostrophe.

Both is fixed below ...

```
re.compile(’Mr\.|[\w\’]+|[\.?]’).findall(text)
```

OUTPUT:

```
['Mr.',
 'Bob',
 'Dobolina',
 'is',
 "thinkin'",
 'of',
 'a',
 'master',
 'plan',
 '.',
 'Why',
 "doesn't",
 'he',
 'quit',
 '?']
```

OUTPUT:

```
['Mr.',
 'Bob',
 'Dobolina',
 'is',
 "thinkin'",
 'of',
 'a',
 'master',
 'plan',
 '.',
 'Why',
 "doesn't",
 'he',
 'quit',
 '?']
```

- Can you think of a complication when recognizing words instead of splits?

- Finding words instead of splits is risky
- We do not want input to disappear

Is tokenization a solved problem?

- For English simple pattern matching can be sufficient.
- However, be careful with "real" data!
- In other languages (e.g. Japanese), words are not separated by whitespace.

Is tokenization a solved problem?

- For English simple pattern matching can be sufficient.
- However, be careful with "real" data!
- In other languages (e.g. Japanese), words are not separated by whitespace.

tency of the training data. We conclude that besides inconsistencies in the data and exceptional cases the task can be considered solved for Latin languages (>99.5 F1). However, performance is 0.75 F1 point lower on average for datasets in other scripts and performance deteriorates in cross-dataset setups.

```
jap = "今日もしないといけない。"
```

```
jap = "今日もしないといけない。"
```

Try lexicon-based tokenization ...

```
jap = "今日もしないといけない。"
```

Try lexicon-based tokenization ...

```
re.compile('もし|今日|も|しない|と|けない').findall(jap)
```

OUTPUT:

['今日', 'もし', 'と', 'けない']

Equally complex for certain English domains (eg. bio-medical text).

```
bio = """We developed a nanocarrier system of
herceptin-conjugated nanoparticles of
d-alpha-tocopheryl-co-poly(ethylene glycol) 1000
succinate (TPGS)-cisplatin prodrug ..."""
```

```
bio = """We developed a nanocarrier system of
herceptin-conjugated nanoparticles of
d-alpha-tocopheryl-co-poly(ethylene glycol) 1000
succinate (TPGS)-cisplatin prodrug ..."""
```

- d-alpha-tocopheryl-co-poly is one token
- (TPGS)-cisplatin are five:
  - (
  - TPGS
  -)
  - -
  - cisplatin

```
re.compile('\s').split(bio)[:15]
```

OUTPUT:

```
['We',
 'developed',
 'a',
 'nanocarrier',
 'system',
 'of',
 'herceptin-conjugated',
 'nanoparticles',
 'of',
 'd-alpha-tocopheryl-co-poly(ethylene',
 'glycol)',
 '1000',
 'succinate',
 '(TPGS)-cisplatin',
 'prodrug']
```

OUTPUT:

```
['We',
 'developed',
 'a',
 'nanocarrier',
 'system',
 'of',
 'herceptin-conjugated',
 'nanoparticles',
 'of',
 'd-alpha-tocopheryl-co-poly(ethylene',
 'glycol)',
 '1000',
 'succinate',
 '(TPGS)-cisplatin',
 'prodrug']
```

www.menti.com: 451093

Labs

Thanks for today!