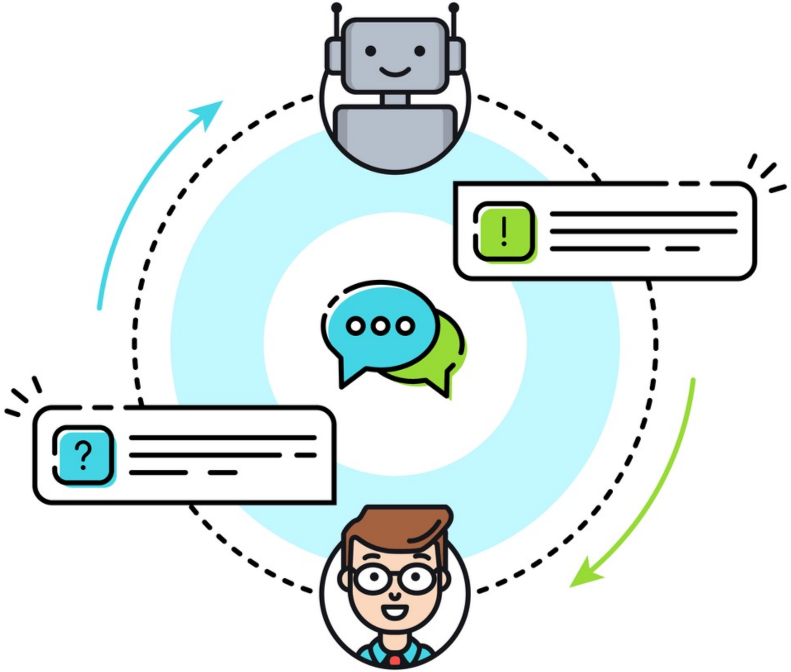
Natural Language Processing

Week 1: Introduction to NLP

Natural Language Processing

- **Definition**: A field that addresses methods and approaches with which computers can process, understand and generate natural language

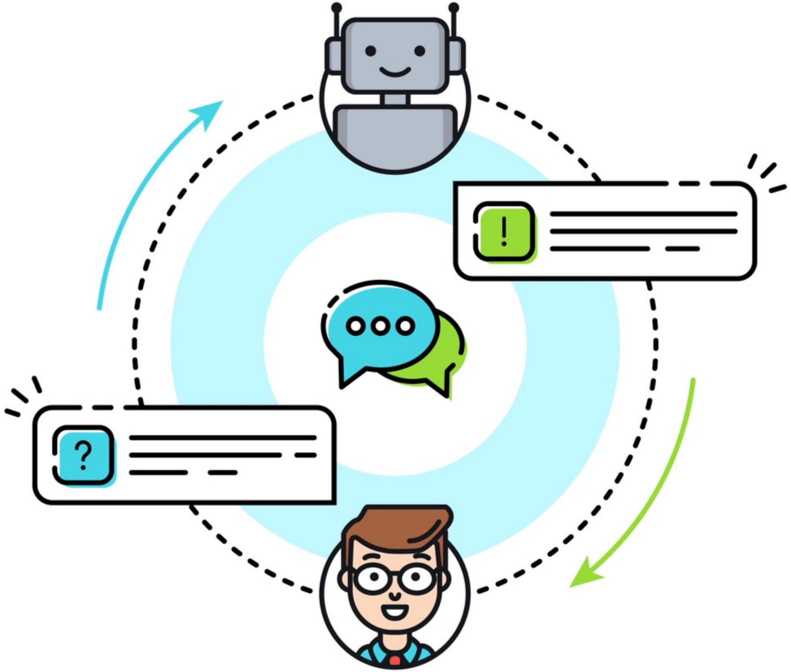
- **Why study NLP?** We use language to read, write and speak, but also: to make plans and decisions, to learn, to dream, and much more. If we want to build intelligent systems, they should have similar abilities



Natural Language Processing

- **Who is this course for?** Anyone working in CS, but especially interested in ML and AI. You are already familiar with many NLP applications. You are already an “expert” in language tasks

- **What will we cover?** A wide range of NLP techniques and applications; theoretical knowledge + practical skills. By the end of the course, you’ll be able to implement your own NLP project end-to-end



This week

01 Overview of NLP applications

02 Building blocks of NLP applications

03 Implementation of a simple NLP application

This week

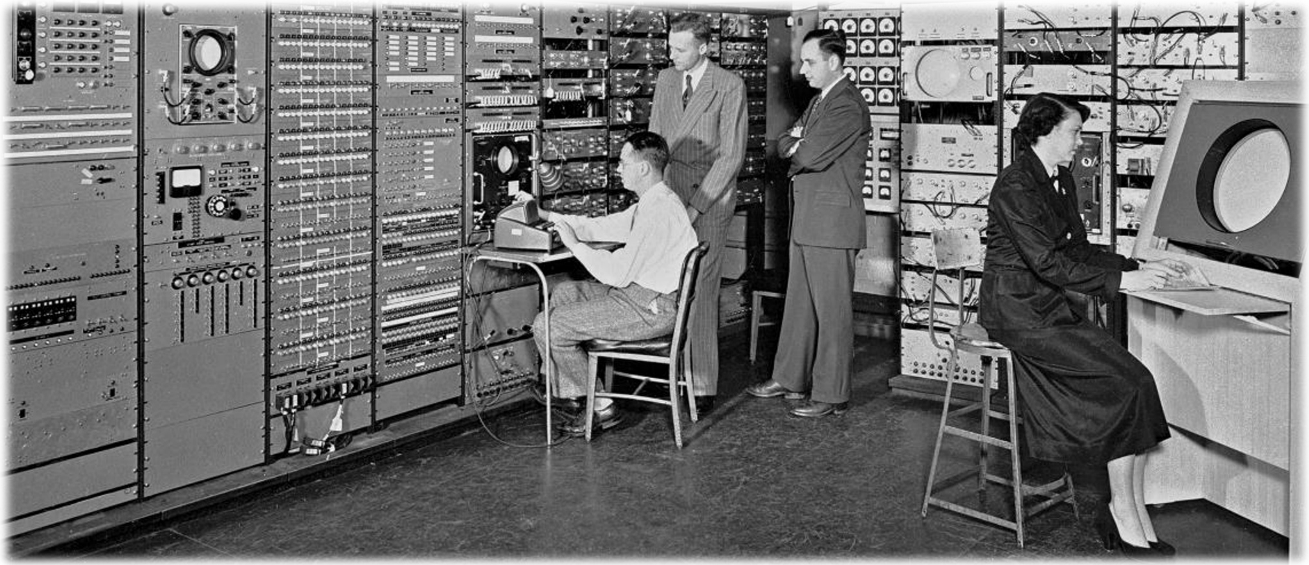
04 Introduction to text tokenization

05 Introduction to linguistic analysis

06 Course logistics

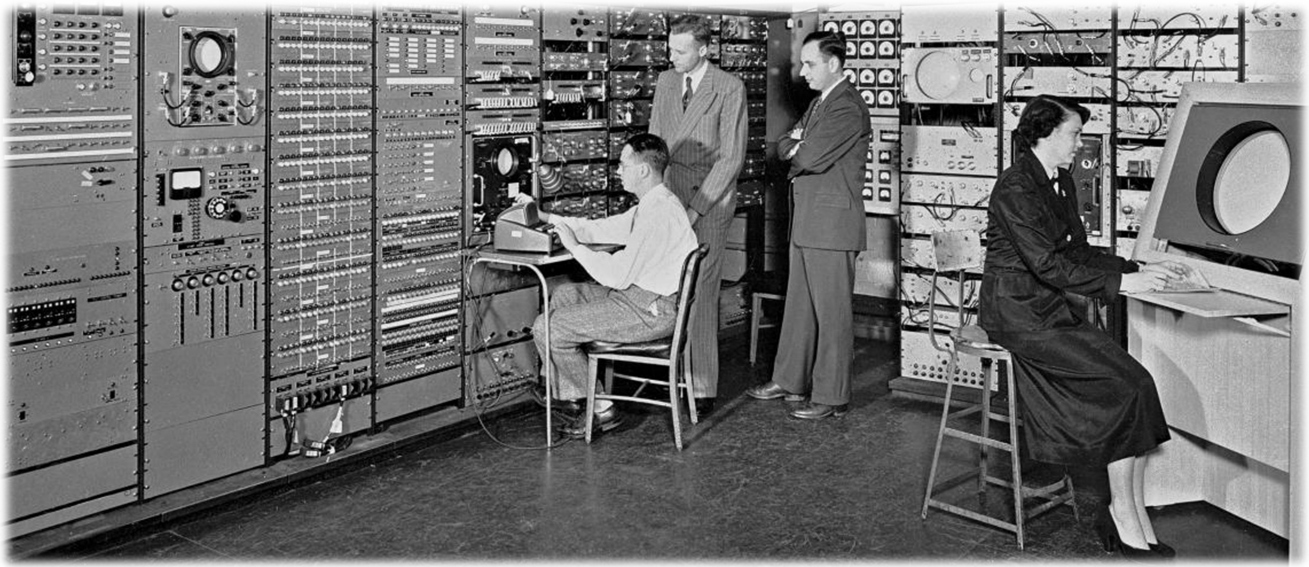
Overview of NLP applications

A bit of history



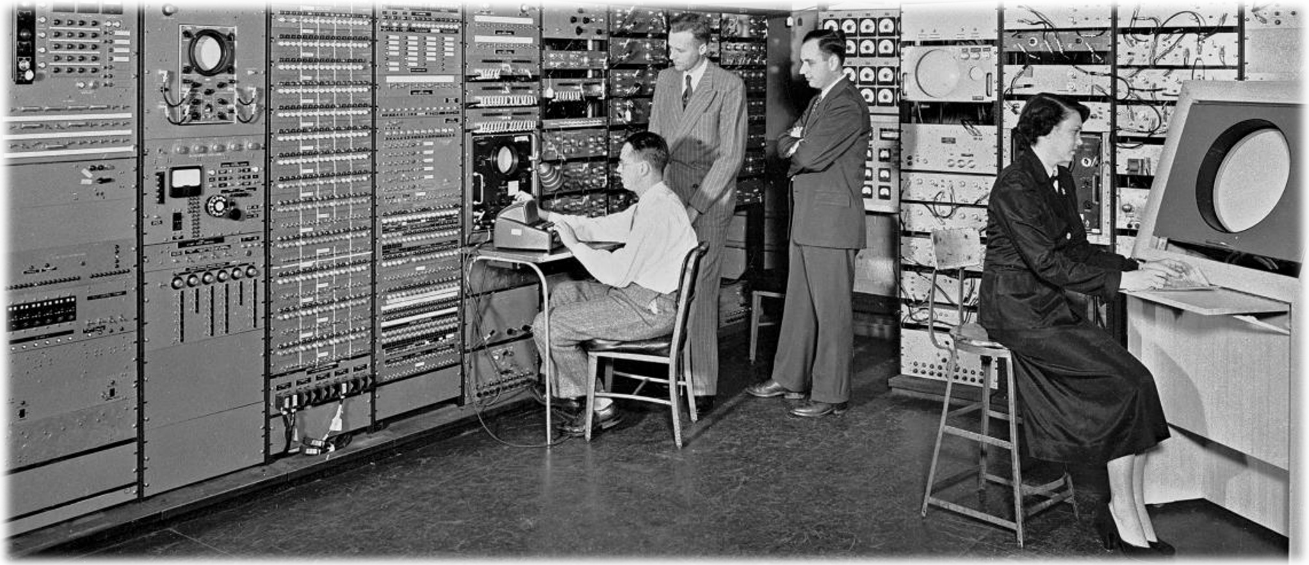
- The field was established in **1950s** and started with the Georgetown-IBM experiment
- The experiment was concerned with the implementation of an early fully-automated **machine translation** system

A bit of history



- Specifically interested in translating between **Russian and English** scientific text
- The task was deemed to be **easy enough** to be solved within several years

A bit of history



Do you think they succeeded? Why?

Development of approaches

- Started off with **rule-based approaches** and **templates**:
    - How do you think it would work for a machine translation system?

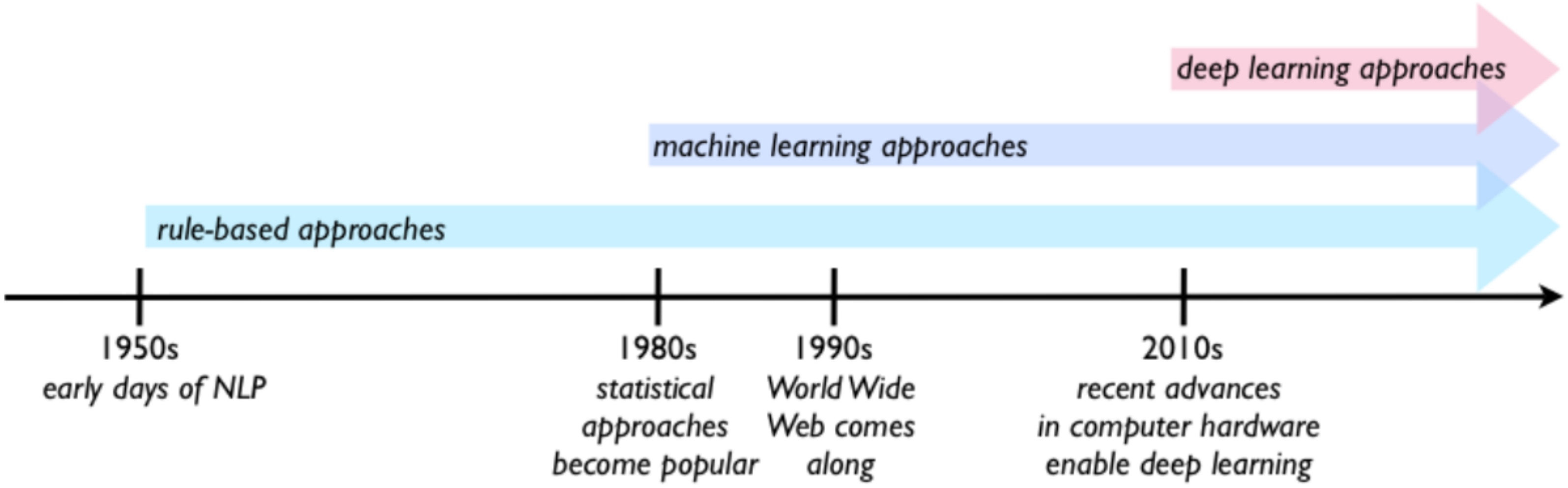
Development of approaches

- Started off with **rule-based approaches** and **templates**:
    - For a machine translation system, we try to translate word for word. Do you think this works well?
- Around 1980s, **statistical approaches** were introduced, and **machine learning algorithms** were developed:
    - **Pros**: do not make rigid assumptions and can learn flexibly
    - **Cons**: rely on availability of large amounts of high-quality representative data

Development of approaches

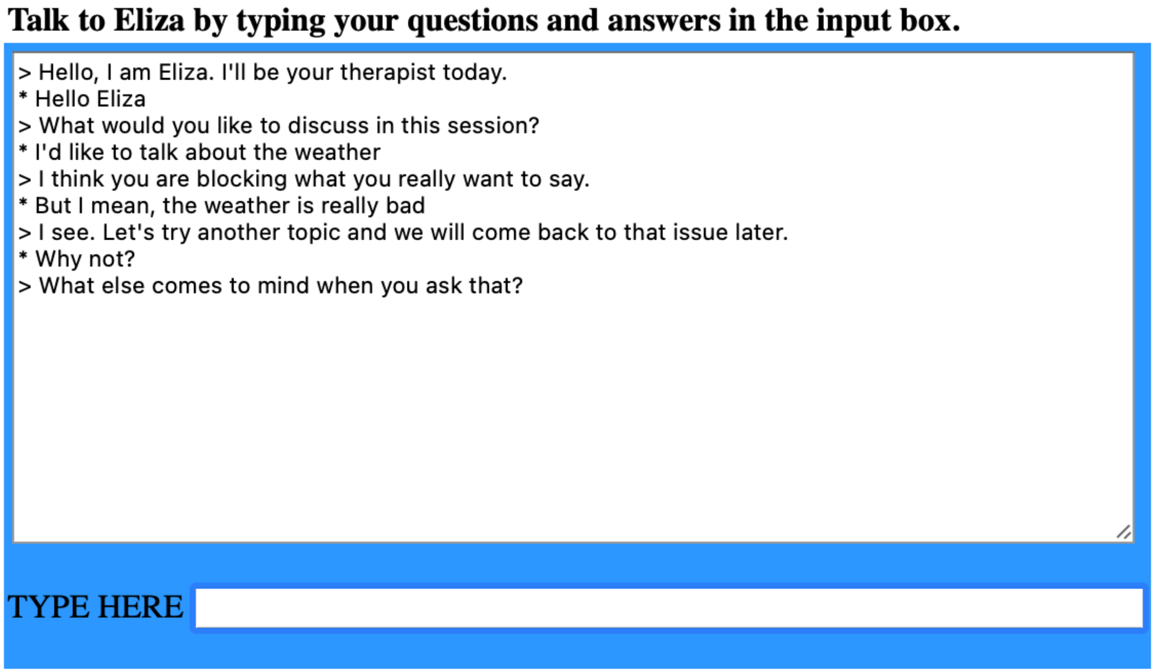
- Started off with **rule-based approaches** and **templates**:
  - For a machine translation system, we try to translate word for word. Do you think this works well?
- Around 1980s, **statistical approaches** were introduced, and **machine learning algorithms** were developed:
  - **Pros**: do not make rigid assumptions and can learn flexibly
  - **Cons**: rely on availability of large amounts of high-quality representative data
- Around 2010s, advances in compute power allowed researchers to apply **deep learning techniques**

NLP timeline


deep learning approaches
machine learning approaches
rule-based approaches
1950s
early days of NLP
1980s
statistical approaches become popular
1990s
World Wide Web comes along
2010s
recent advances in computer hardware enable deep learning

- Note that it does not mean that DL algorithms are the solution to all problems
- Different tasks use different types of solutions, including rule-based approaches

Example: ELIZA chatbot



Talk to Eliza by typing your questions and answers in the input box.

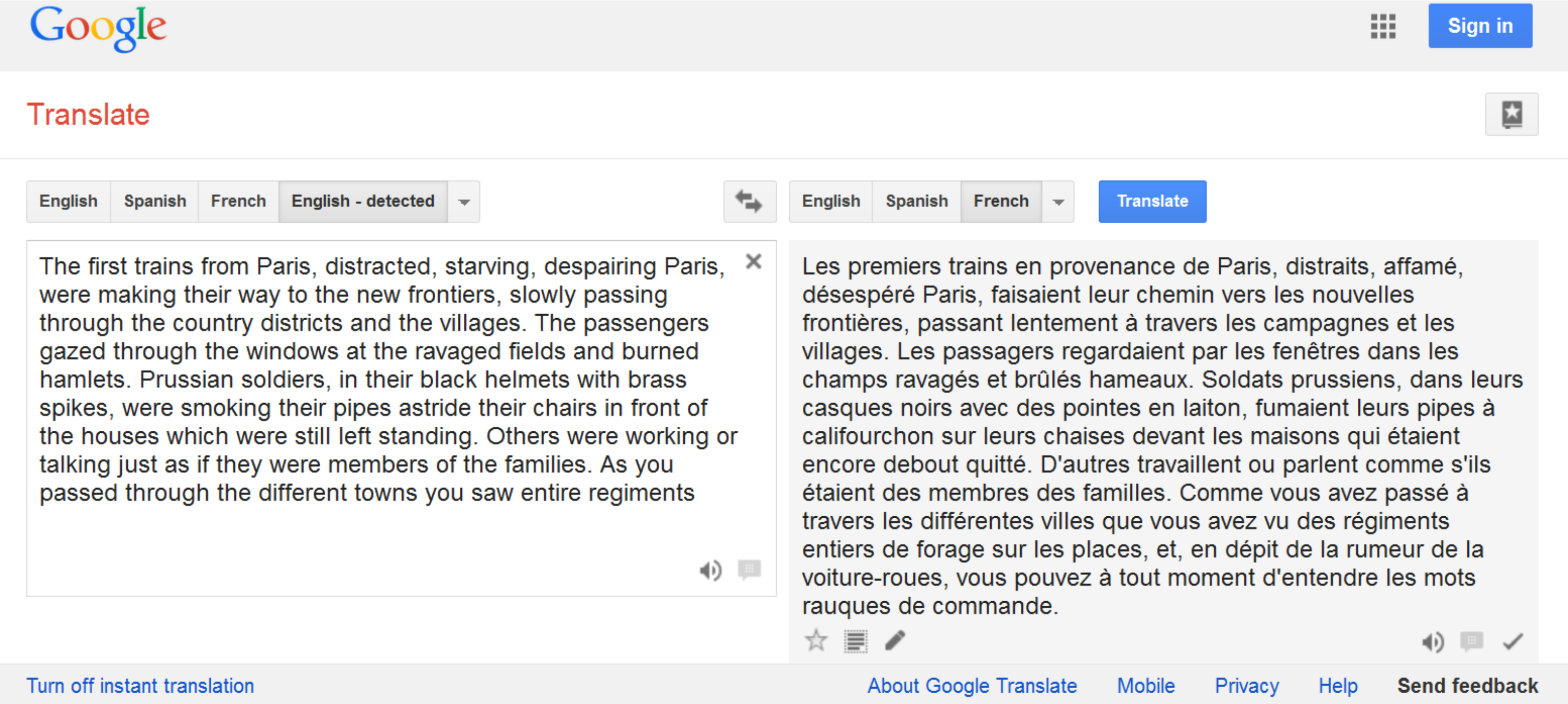
```
> Hello, I am Eliza. I'll be your therapist today.
* Hello Eliza
> What would you like to discuss in this session?
* I'd like to talk about the weather
> I think you are blocking what you really want to say.
* But I mean, the weather is really bad
> I see. Let's try another topic and we will come back to that issue later.
* Why not?
> What else comes to mind when you ask that?
```

- Works by application of templates to “parrot” back what the user is saying

https://web.njit.edu/~ronkowit/eliza.html

What NLP applications do you know of?

Machine Translation: Task



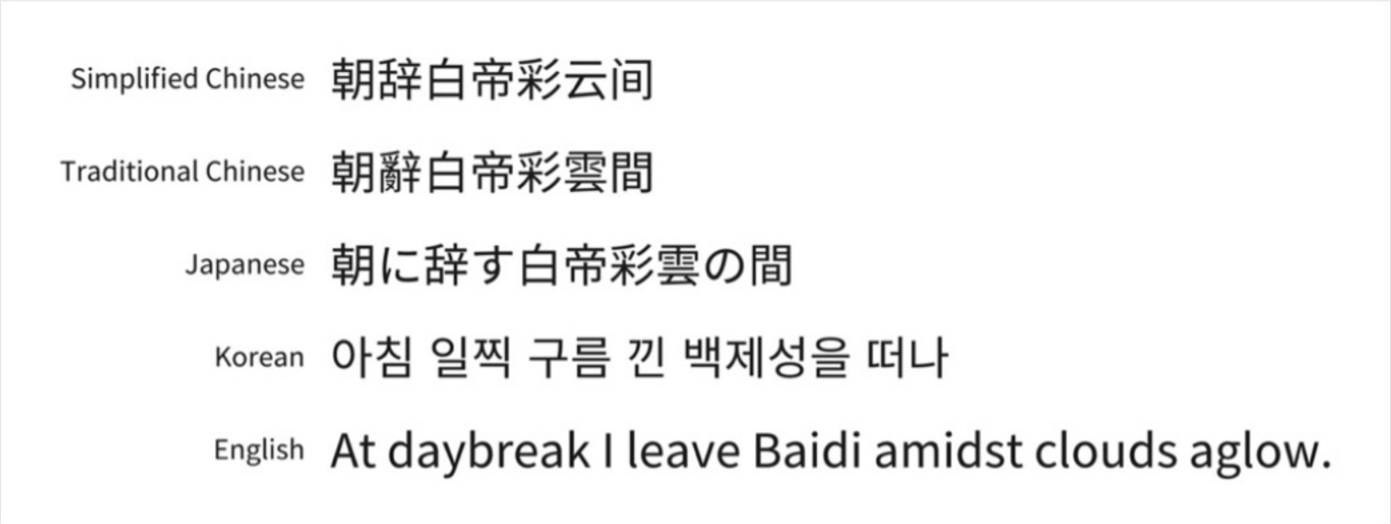
The first trains from Paris, distracted, starving, despairing Paris, were making their way to the new frontiers, slowly passing through the country districts and the villages. The passengers gazed through the windows at the ravaged fields and burned hamlets. Prussian soldiers, in their black helmets with brass spikes, were smoking their pipes astride their chairs in front of the houses which were still left standing. Others were working or talking just as if they were members of the families. As you passed through the different towns you saw entire regiments

Les premiers trains en provenance de Paris, distraits, affamé, désespéré Paris, faisaient leur chemin vers les nouvelles frontières, passant lentement à travers les campagnes et les villages. Les passagers regardaient par les fenêtres dans les champs ravagés et brûlés hameaux. Soldats prussiens, dans leurs casques noirs avec des pointes en laiton, fumaient leurs pipes à califourchon sur leurs chaises devant les maisons qui étaient encore debout quitté. D'autres travaillent ou parlent comme s'ils étaient des membres des familles. Comme vous avez passé à travers les différentes villes que vous avez vu des régiments entiers de forage sur les places, et, en dépit de la rumeur de la voiture-roues, vous pouvez à tout moment d'entendre les mots rauques de commande.

Machine Translation: Challenges

- Human language is creative ⇒ it’s impossible to come up with a generalizable set of rules
- What is a **word**?
  - *Kraftfahrzeug-Haftpflichtversicherung* (DE) = motor-vehicle-liability-insurance (EN)
  - *Auf Wiedersehen* (DE) = goodbye (EN), where *Wiedersehen* = *see again*

Where things get even more complicated



Simplified Chinese	朝辞白帝彩云间
Traditional Chinese	朝辭白帝彩雲間
Japanese	朝に辞す白帝彩雲の間
Korean	아침 일찍 구름 낀 백제성을 떠나
English	At daybreak I leave Baidi amidst clouds aglow.

Machine Translation: Challenges

- Human language is creative ⇒ it’s impossible to come up with a generalizable set of rules
- What is a **word**?
- Different **grammatical categories**:
    - *langage naturel* (FR) = *natural language* (EN)
    - *catastrophe naturelle* (FR) = *natural disaster* (EN)
    - *ressources naturelles* (FR) = *natural resources* (EN)

Machine Translation: Challenges

- Human language is creative ⇒ it’s impossible to come up with a generalizable set of rules
- What is a **word**?
- Different **grammatical categories**
- Different **word order**
    - EN: I$_{\text{SUBJECT}}$ read$_{\text{VERB}}$ a book$_{\text{OBJECT}}$ ⇒ **SVO**
    - JA: I$_{\text{SUBJECT}}$ a book$_{\text{OBJECT}}$ read$_{\text{VERB}}$ ⇒ **SOV**
    - GA (Irish): read$_{\text{VERB}}$ I$_{\text{SUBJECT}}$ a book$_{\text{OBJECT}}$ ⇒ **VSO**
    - RU: **Flexible** in terms of word order

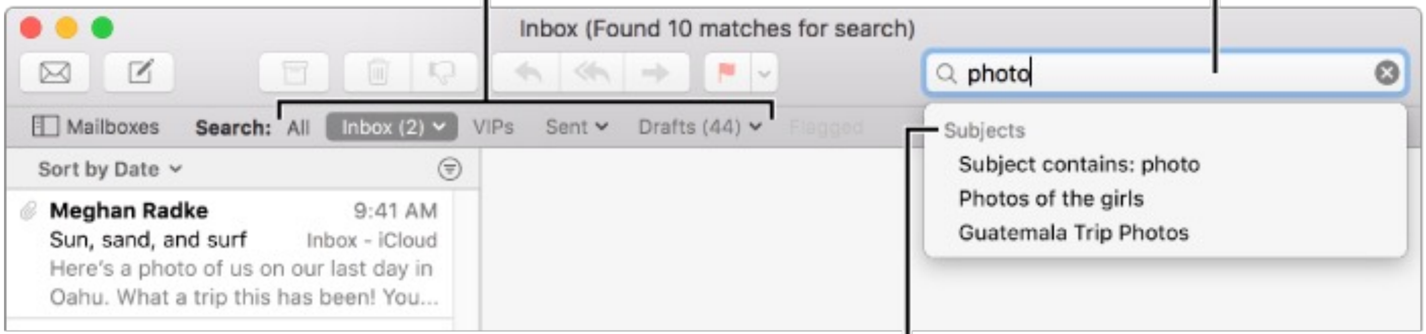
Machine Translation: Challenges

- Human language is creative ⇒ it’s impossible to come up with a generalizable set of rules
- What is a **word**?
- Different **grammatical categories**
- Different **word order**
- Words that only appear to **mean** the same thing:
    - *I’ll **book** a trip ≠ I like this **book***

Information Search: Task

Click All or a different mailbox to change results.

Type or paste text. Or drag an email address from a message.



Inbox (Found 10 matches for search)

photo

Meghan Radke 9:41 AM

Sun, sand, and surf Inbox - iCloud

Here's a photo of us on our last day in Oahu. What a trip this has been! You...

Subjects

Subject contains: photo

Photos of the girls

Guatemala Trip Photos

Suggestion categories vary based on your search text.

Information Search: Challenges

- Suppose you have **thousands of documents**, each containing hundreds of pages
- Suppose also that you are looking for information on a **specific concept**, e.g., *reinforcement learning*
- How would you perform such search?
- How would you judge if the results are relevant?

Information Search: Challenges

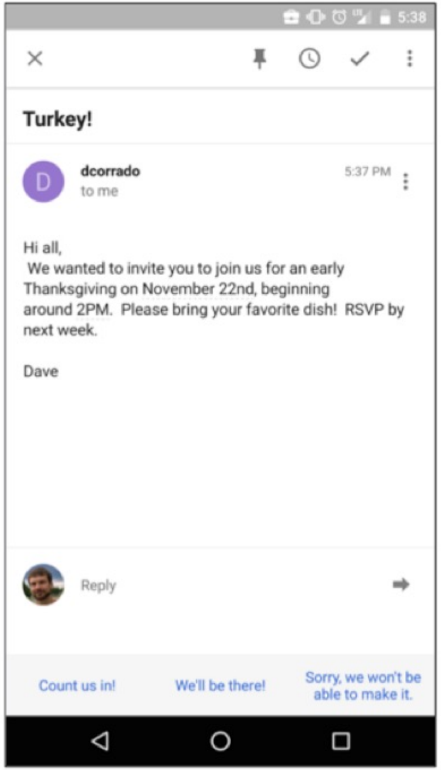
- Queries may be **incomplete**, **inaccurate**, **ungrammatical**, etc.
- They may also be **ambiguous**: when you type in “book”, what do you mean?
- Some words **matter more** than others:
  - *I’m looking for accommodation in Bath*
- Some words mean **similar** things:
  - *I’m looking for accommodation in Bath ⇔ There are flats in Widcombe*

Spam Filtering: Task & Challenges

- Identification of a potentially malicious, unsafe or dangerous content
- **Why is this challenging?** Some emails may contain clear "red flags" (unusual formatting, unknown sender, mass emailing, etc.); others can only be identified by their content

Text prediction: Task & Challenges

- Widely used in predictive keyboards on smartphones, in browsers, email clients (e.g., Smart Reply), etc.
- The most likely continuation is suggested based on the beginning of a word or a phrase



5:38

Turkey!

dcorrado
to me

5:37 PM

Hi all,
We wanted to invite you to join us for an early Thanksgiving on November 22nd, beginning around 2PM. Please bring your favorite dish! RSVP by next week.

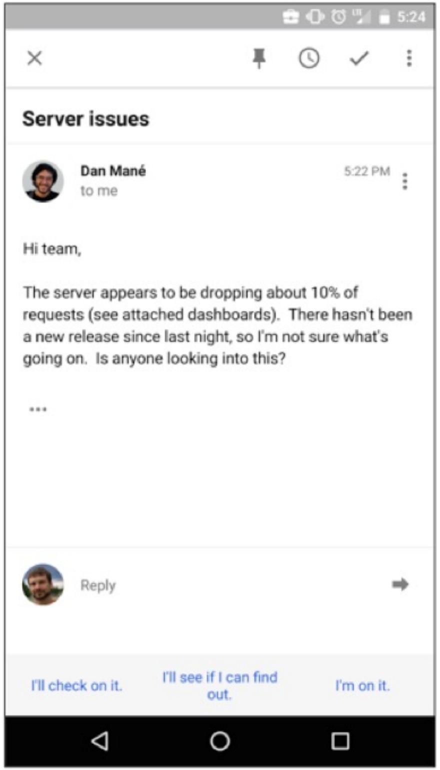
Dave

Reply

Count us in!

We'll be there!

Sorry, we won't be able to make it.



5:24

Server issues

Dan Mané
to me

5:22 PM

Hi team,

The server appears to be dropping about 10% of requests (see attached dashboards). There hasn't been a new release since last night, so I'm not sure what's going on. Is anyone looking into this?

...

Reply

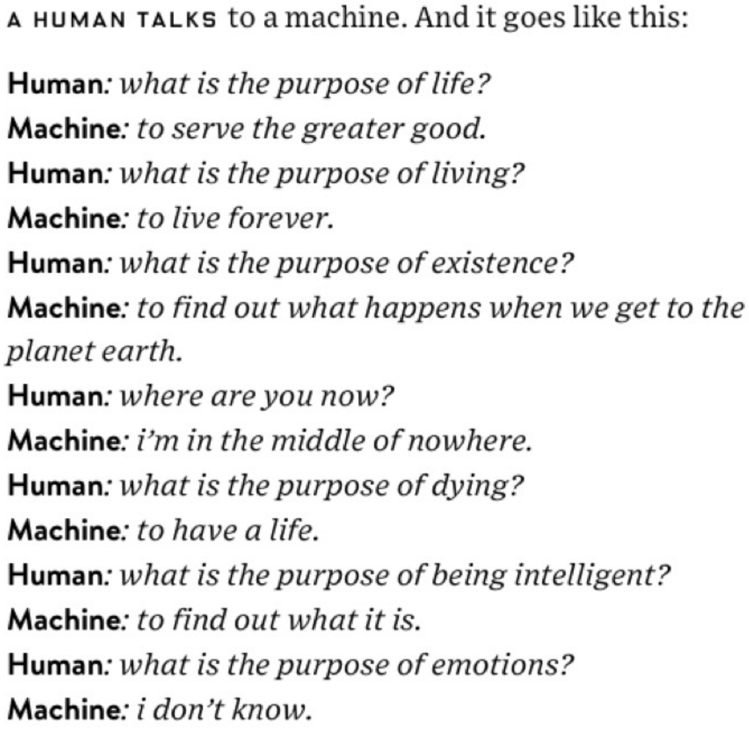
I'll check on it.

I'll see if I can find out.

I'm on it.

Text prediction: Task & Challenges

- Google's Smart Reply may even compose whole (short) emails like "Monday works for me" or "Sounds good" on your behalf
- **Why is this challenging?** The range of possible natural sentences is practically infinite. An ability to predict what comes next in language brings machine intelligence one step closer to human intelligence



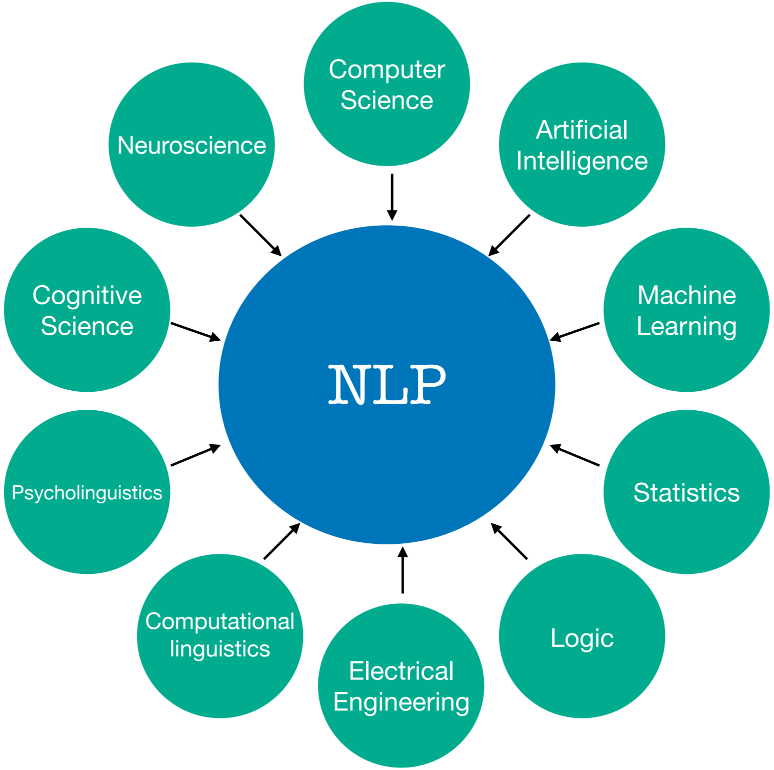
A HUMAN TALKS to a machine. And it goes like this:

Human: *what is the purpose of life?*
Machine: *to serve the greater good.*
Human: *what is the purpose of living?*
Machine: *to live forever.*
Human: *what is the purpose of existence?*
Machine: *to find out what happens when we get to the planet earth.*
Human: *where are you now?*
Machine: *i'm in the middle of nowhere.*
Human: *what is the purpose of dying?*
Machine: *to have a life.*
Human: *what is the purpose of being intelligent?*
Machine: *to find out what it is.*
Human: *what is the purpose of emotions?*
Machine: *i don't know.*

http://www.wired.com/2015/06/google-made-chatbot-debates-meaning-life

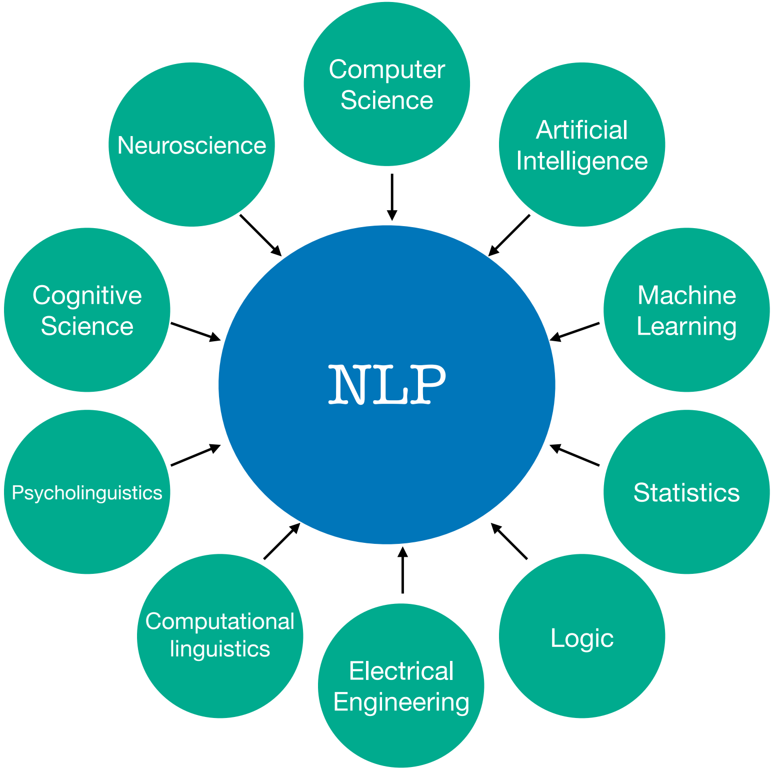
NLP in relation to other fields

- **Computer Science** contributes algorithms, software & hardware
- **Artificial Intelligence** sets up the environment for intelligent machines
- **Machine Learning** algorithms are widely used in NLP
- **Statistics** helps coming up with theoretical models and probabilistic interpretation of language phenomena



NLP in relation to other fields

- **Logic** helps making sure the world described with NLP models makes sense
- **Electrical engineering** techniques help with specific tasks (e.g., speech processing)
- **Computational linguistics** provides expert knowledge about how language works
- **Other fields** account for human factors, brain processes, etc.



Building blocks of NLP applications

Concepts and methods

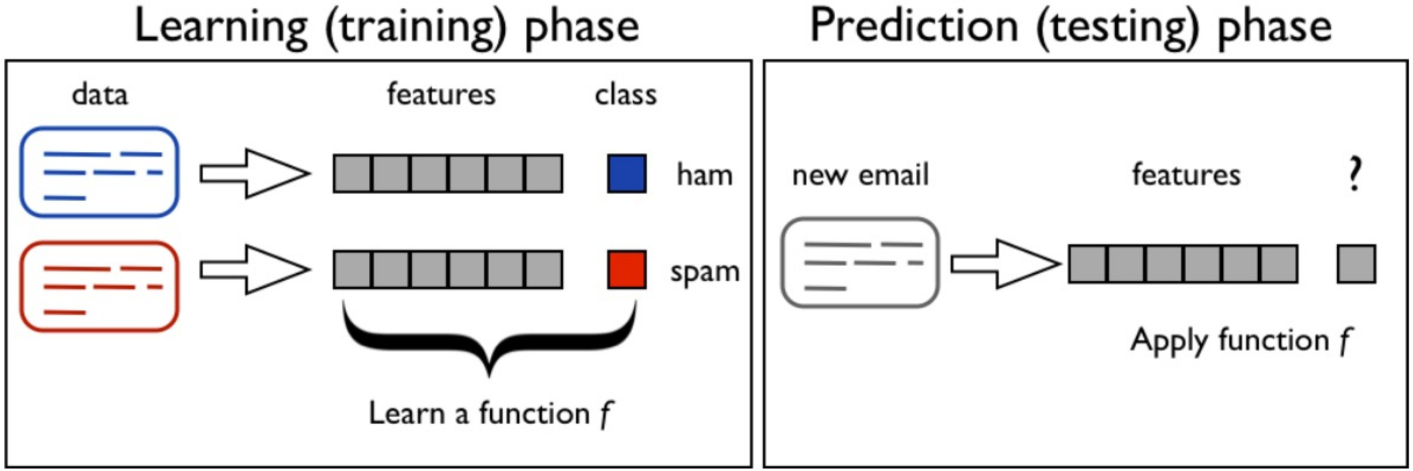
- **Machine learning methods** are widely used
- If the goal is to predict a label selecting among a finite set of categories, this is *classification* or *categorization.* Can you think of examples?

Concepts and methods

- **Machine learning methods** are widely used
- If the goal is to predict a label selecting among a finite set of categories, this is *classification* or *categorization.* Can you think of examples?
    - spam filtering – binary (spam / ham)
    - topic classification – multi-class (finance / sports / arts / science)

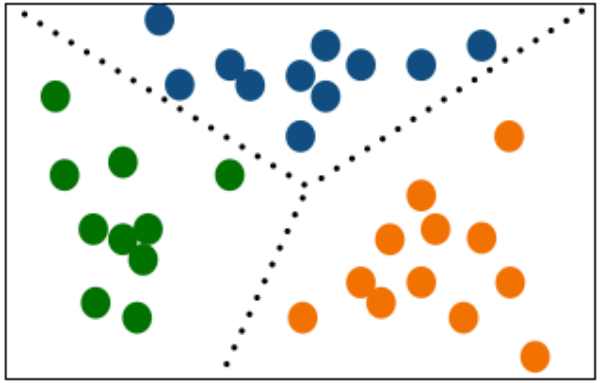
Concepts and methods

Often annotated datasets are available in open access for such purposes. If you have labelled data, you can apply **supervised machine learning** techniques: the ML algorithm tries to lean a function mapping the characteristics of the data from each class or category to the respective label



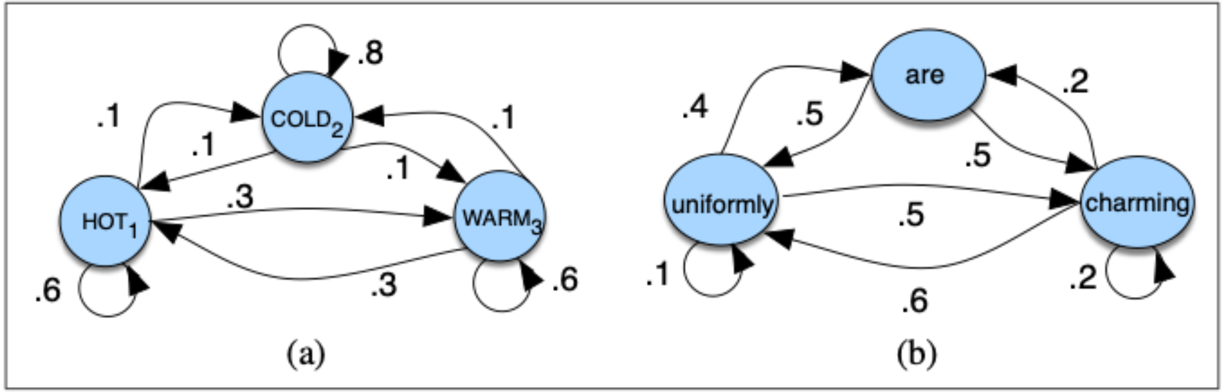
Concepts and methods

- Data labelling is expensive and time-consuming, so labelled data is not always readily available
- If labelled data is not available, **unsupervised machine learning** approaches can be used: e.g., *clustering* to identify groups of similar documents based on their content or *Latent Dirichlet Allocation* (LDA) used to detect topics in unlabelled data



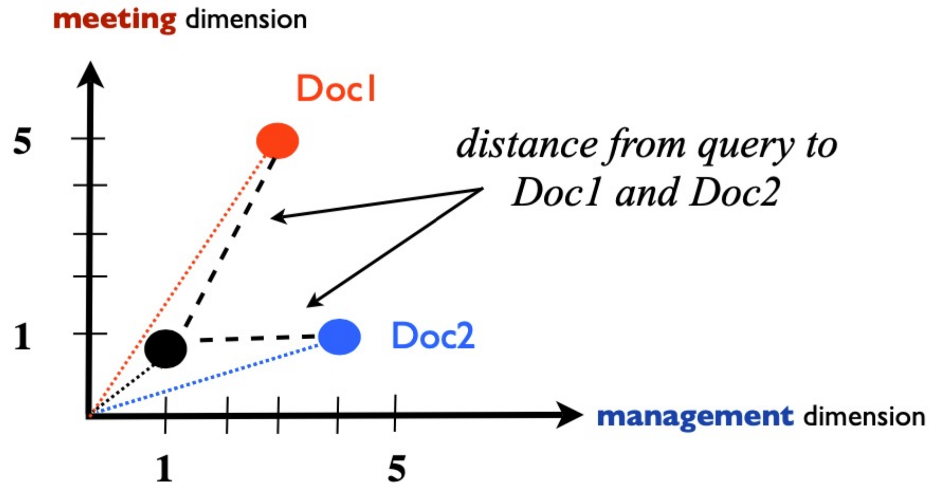
Concepts and methods

- Certain language phenomena are best described as sequences of events rather than as individual occurrences: e.g., the way characters follow each other in a word or words follow each other in a sentence is not random
- **Sequence modelling approaches** help to address such tasks as *part-of-speech tagging* and *language modelling*, among others



Concepts and methods

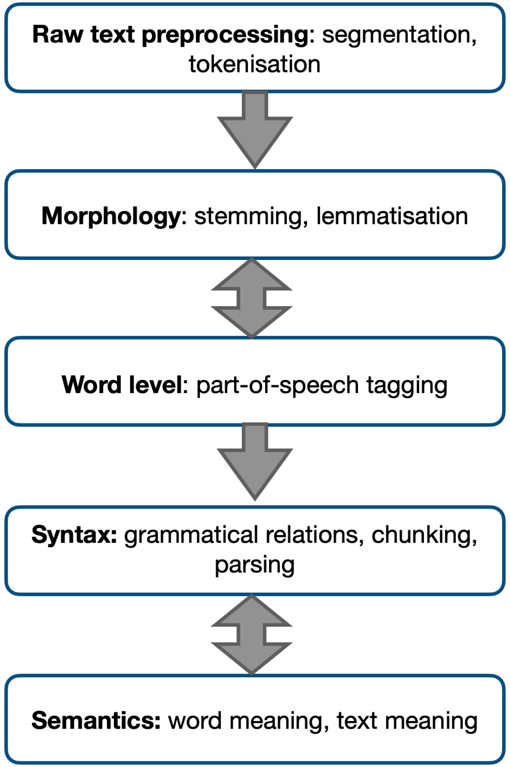
- Finally, a number of NLP applications rely on **vector-based models**
- Such vectors may encode word occurrences across documents (as in information retrieval) or aspects of meaning across the vocabulary (as in semantic models and word vectors / embeddings)



Levels of linguistic analysis

- **Raw text processing**: Note that a computer doesn’t have an idea of what a “word” is – text is a single stream of symbols. How should we split this stream into units?

- **Morphology**: Sub-word level of linguistic analysis
    - *book* (singular) vs. *books* (plural)
    - *(will) book (tomorrow)* vs. *(is) booking (now)* vs. *booked (yesterday)*
    - *(I) book* vs. *(he) books*
    - More challenging still: *am / is / are / was / were / been / …* = *be*



Raw text preprocessing: segmentation, tokenisation

Morphology: stemming, lemmatisation

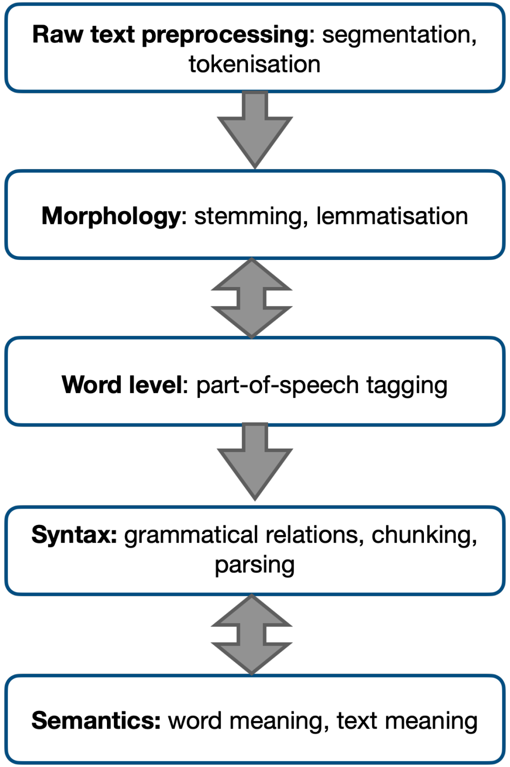
Word level: part-of-speech tagging

Syntax: grammatical relations, chunking, parsing

Semantics: word meaning, text meaning

Levels of linguistic analysis

- **Word level analysis**: Identification of word types – *parts of speech*. E.g., *interesting book* (noun) vs *to book a ticket* (verb)
- **Syntax**: deals with the way words are grouped together in sentences to express meaning. E.g., *Police help dog bite victim* – who did what to whom?
- **Semantics**: addresses questions related to the meaning of words and other language units. E.g., *plot of land ≠ plot of a story*


Raw text preprocessing: segmentation, tokenisation

Morphology: stemming, lemmatisation

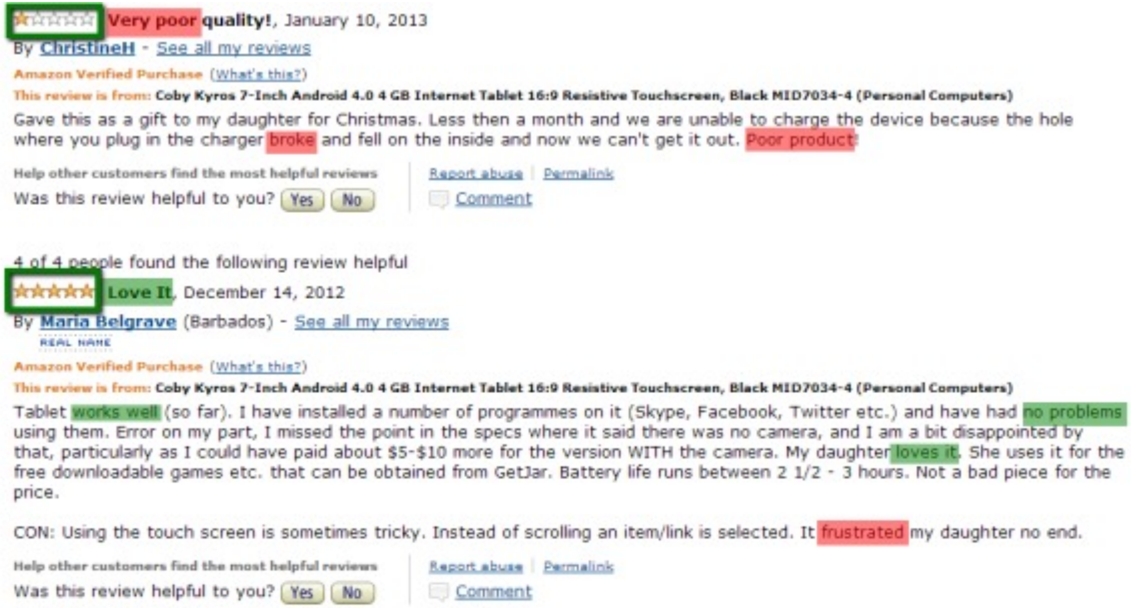
Word level: part-of-speech tagging

Syntax: grammatical relations, chunking, parsing

Semantics: word meaning, text meaning

Quiz

Sentiment analysis is the task concerned with automated identification of the polarity of the opinion expressed in text: e.g., positive or negative sentiment in reviews



★☆☆☆☆ **Very poor quality!**, January 10, 2013

By **ChristineH** - See all my reviews

Amazon Verified Purchase (What's this?)

This review is from: **Coby Kyros 7-Inch Android 4.0 4 GB Internet Tablet 16:9 Resistive Touchscreen, Black MID7034-4 (Personal Computers)**

Gave this as a gift to my daughter for Christmas. Less then a month and we are unable to charge the device because the hole where you plug in the charger broke and fell on the inside and now we can't get it out. Poor product!

Help other customers find the most helpful reviews

Was this review helpful to you? Yes No

Report abuse | Permalink

Comment

4 of 4 people found the following review helpful

★★★★★ **Love It**, December 14, 2012

By **Maria Belgrave** (Barbados) - See all my reviews

REAL NAME

Amazon Verified Purchase (What's this?)

This review is from: **Coby Kyros 7-Inch Android 4.0 4 GB Internet Tablet 16:9 Resistive Touchscreen, Black MID7034-4 (Personal Computers)**

Tablet works well (so far). I have installed a number of programmes on it (Skype, Facebook, Twitter etc.) and have had no problems using them. Error on my part, I missed the point in the specs where it said there was no camera, and I am a bit disappointed by that, particularly as I could have paid about $5-$10 more for the version WITH the camera. My daughter loves it. She uses it for the free downloadable games etc. that can be obtained from GetJar. Battery life runs between 2 1/2 - 3 hours. Not a bad piece for the price.

CON: Using the touch screen is sometimes tricky. Instead of scrolling an item/link is selected. It frustrated my daughter no end.

Help other customers find the most helpful reviews

Was this review helpful to you? Yes No

Report abuse | Permalink

Comment

Quiz

Which options apply?

A. This is a **sequence labelling task**, as the sentiment in each text depends on the sentiments in the previous texts.

B. This task can benefit from **various types of linguistic analysis**, from word-level (e.g., identification of individual sentiment-bearing words) to syntactic (identification of groups of such words) to semantic (meaning analysis).

C. There is **no use for syntactic analysis** in this application: extraction of single words should be enough as one can detect sentiment of text based on the sentiments expressed by individual words.

D. This task can be solved using a **binary classification algorithm** to distinguish between the two classes – positive and negative sentiment.

Quiz

Which options apply?

A. This is a **sequence labelling task**, as the sentiment in each text depends on the sentiments in the previous texts.

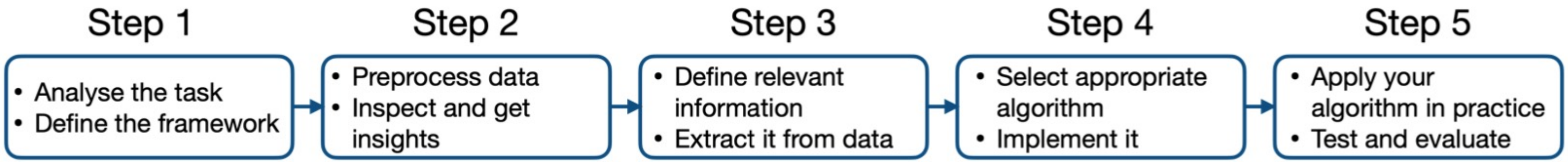
B. This task can benefit from **various types of linguistic analysis**, from word-level (e.g., identification of individual sentiment-bearing words) to syntactic (identification of groups of such words) to semantic (meaning analysis).

C. There is **no use for syntactic analysis** in this application: extraction of single words should be enough as one can detect sentiment of text based on the sentiments expressed by individual words.

D. This task can be solved using a **binary classification algorithm** to distinguish between the two classes – positive and negative sentiment.

Implementation of a simple NLP application

Pipeline overview



Step 1

- Analyse the task
- Define the framework

Step 2

- Preprocess data
- Inspect and get insights

Step 3

- Define relevant information
- Extract it from data

Step 4

- Select appropriate algorithm
- Implement it

Step 5

- Apply your algorithm in practice
- Test and evaluate

- Any task can be thought of in terms of this pipeline
- Let’s walk through these five steps for **spam filtering**

Step 1: Analysis of the task

- Define **what exactly the task involves**: e.g., ask yourself, how you would solve it yourself (without ML)
- In **spam filtering**: you probably pay attention to certain characteristics (sender, fonts, format, how many recipients the email has, etc.)
- You also may pay attention to the **content**: “lottery”, “click on this link”, “your account is blocked”, and similar
- Most probably, you **classify** the emails in two types – normal emails and spam
- ⇒ **Binary classification task**

Step 2: Analysis and preprocessing of the data

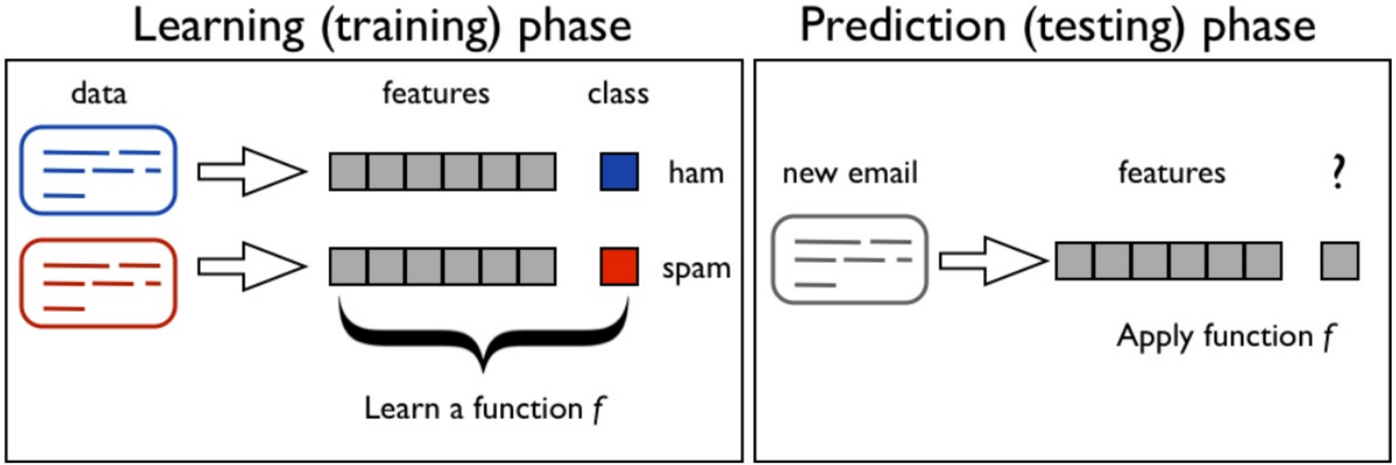
- Given the “red flags” (words and phrases) you may attempt using **templates**
- For machine learning, define what the **relevant data** is and how to **prepare it**:
    - You need access to labelled data of two classes
    - What is the distribution of classes?
    - Are you going to use only textual features?
    - Are there any other significant differences (e.g., spam emails being considerably shorter)?

Step 3: Definition and extraction of the relevant information

- Identify **relevant signal in the data**
    - Is it single words (“lottery”, “blocked”) or phrases (“click on this link”)?
    - Are you going to learn from misspellings?
    - Are you going to learn from different ways to spell words (e.g., “Now”, “now”, “NOW”)?
    - Are you going to learn from word occurrences or word distribution?
    - Will you apply any other normalisation techniques?
- The above points refer to **feature selection**, **feature representation**, and **feature weighting**

Step 4: Implementation of the algorithm

- No algorithm can be considered absolutely the best for all tasks and all datasets (“no free lunch theorem”)
- Analyse the task to identify which one suits best in each particular case



Step 5: Testing and evaluation

- It is important to understand how your current algorithm performs and what you can do better
- E.g., for classification tasks, you can measure accuracy, precision, recall, F1
- Arguably, it is better to let some annoying spam messages to slip through than send important “normal” emails to the spam box – is **precision or recall** more important?
- It is advisable to set up some **baseline**: What is the majority class distribution? How would the simplest algorithm perform? Are you really doing better using a more sophisticated approach?

Quiz

- Suppose you are developing an email application that can not only distinguish between spam and ham, but also **detect more fine-grained categories** in your emails (e.g., “work-related”, “personal”, “notifications”, etc.)
- You are provided with the **data labelled** with these categories to help you build this application

What steps will be involved in the development of this application? Select all that apply and put them in the right order.

a	I will consider using feature normalisation techniques.
b	I will deploy my algorithm in practice, for example, will put it on GitHub and encourage people to use it.
c	I will focus specifically on the implementation and application of the Naïve Bayes algorithm.
d	I will inspect the data to see what it contains.
e	I will build a binary classification algorithm.
f	I will write some rules based on the keywords: for example, if an email contains a word “work”, it should be assigned with the “work-related” category, and so on.
g	I will implement and apply an appropriate classification algorithm capable of dealing with multiple classes.
h	I will thoroughly evaluate my algorithm, compare the results to the existing baselines, and consider possible improvements.
i	I will analyse the task and consider how I would solve it myself.
j	I will select features (e.g., words, word combinations, grammatical patterns) for the machine learning algorithm and extract them from the data.

1 (i)	I will analyse the task and consider how I would solve it myself.
a	I will consider using feature normalisation techniques.
b	I will deploy my algorithm in practice, for example, will put it on GitHub and encourage people to use it.
c	I will focus specifically on the implementation and application of the Naïve Bayes algorithm.
d	I will inspect the data to see what it contains.
e	I will build a binary classification algorithm.
f	I will write some rules based on the keywords: for example, if an email contains a word "work", it should be assigned with the "work-related" category, and so on.
g	I will implement and apply an appropriate classification algorithm capable of dealing with multiple classes.
h	I will thoroughly evaluate my algorithm, compare the results to the existing baselines, and consider possible improvements.
j	I will select features (e.g., words, word combinations, grammatical patterns) for the machine learning algorithm and extract them from the data.

1 (i)	I will analyse the task and consider how I would solve it myself.
2 (d)	I will inspect the data to see what it contains.
a	I will consider using feature normalisation techniques.
b	I will deploy my algorithm in practice, for example, will put it on GitHub and encourage people to use it.
c	I will focus specifically on the implementation and application of the Naïve Bayes algorithm.
e	I will build a binary classification algorithm.
f	I will write some rules based on the keywords: for example, if an email contains a word “work”, it should be assigned with the “work-related” category, and so on.
g	I will implement and apply an appropriate classification algorithm capable of dealing with multiple classes.
h	I will thoroughly evaluate my algorithm, compare the results to the existing baselines, and consider possible improvements.
j	I will select features (e.g., words, word combinations, grammatical patterns) for the machine learning algorithm and extract them from the data.

1 (i)	I will analyse the task and consider how I would solve it myself.
2 (d)	I will inspect the data to see what it contains.
3 (j)	I will select features (e.g., words, word combinations, grammatical patterns) for the machine learning algorithm and extract them from the data.
a	I will consider using feature normalisation techniques.
b	I will deploy my algorithm in practice, for example, will put it on GitHub and encourage people to use it.
c	I will focus specifically on the implementation and application of the Naïve Bayes algorithm.
e	I will build a binary classification algorithm.
f	I will write some rules based on the keywords: for example, if an email contains a word “work”, it should be assigned with the “work-related” category, and so on.
g	I will implement and apply an appropriate classification algorithm capable of dealing with multiple classes.
h	I will thoroughly evaluate my algorithm, compare the results to the existing baselines, and consider possible improvements.

1 (i)	I will analyse the task and consider how I would solve it myself.
2 (d)	I will inspect the data to see what it contains.
3 (j)	I will select features (e.g., words, word combinations, grammatical patterns) for the machine learning algorithm and extract them from the data.
4 (a)	I will consider using feature normalisation techniques.
b	I will deploy my algorithm in practice, for example, will put it on GitHub and encourage people to use it.
c	I will focus specifically on the implementation and application of the Naïve Bayes algorithm.
e	I will build a binary classification algorithm.
f	I will write some rules based on the keywords: for example, if an email contains a word “work”, it should be assigned with the “work-related” category, and so on.
g	I will implement and apply an appropriate classification algorithm capable of dealing with multiple classes.
h	I will thoroughly evaluate my algorithm, compare the results to the existing baselines, and consider possible improvements.

1 (i)	I will analyse the task and consider how I would solve it myself.
2 (d)	I will inspect the data to see what it contains.
3 (j)	I will select features (e.g., words, word combinations, grammatical patterns) for the machine learning algorithm and extract them from the data.
4 (a)	I will consider using feature normalisation techniques.
5 (g)	I will implement and apply an appropriate classification algorithm capable of dealing with multiple classes.
b	I will deploy my algorithm in practice, for example, will put it on GitHub and encourage people to use it.
c	I will focus specifically on the implementation and application of the Naïve Bayes algorithm.
e	I will build a binary classification algorithm.
f	I will write some rules based on the keywords: for example, if an email contains a word “work”, it should be assigned with the “work-related” category, and so on.
h	I will thoroughly evaluate my algorithm, compare the results to the existing baselines, and consider possible improvements.

1 (i)	I will analyse the task and consider how I would solve it myself.
2 (d)	I will inspect the data to see what it contains.
3 (j)	I will select features (e.g., words, word combinations, grammatical patterns) for the machine learning algorithm and extract them from the data.
4 (a)	I will consider using feature normalisation techniques.
5 (g)	I will implement and apply an appropriate classification algorithm capable of dealing with multiple classes.
6 (h)	I will thoroughly evaluate my algorithm, compare the results to the existing baselines, and consider possible improvements.
b	I will deploy my algorithm in practice, for example, will put it on GitHub and encourage people to use it.
c	I will focus specifically on the implementation and application of the Naïve Bayes algorithm.
e	I will build a binary classification algorithm.
f	I will write some rules based on the keywords: for example, if an email contains a word “work”, it should be assigned with the “work-related” category, and so on.

1 (i)	I will analyse the task and consider how I would solve it myself.
2 (d)	I will inspect the data to see what it contains.
3 (j)	I will select features (e.g., words, word combinations, grammatical patterns) for the machine learning algorithm and extract them from the data.
4 (a)	I will consider using feature normalisation techniques.
5 (g)	I will implement and apply an appropriate classification algorithm capable of dealing with multiple classes.
6 (h)	I will thoroughly evaluate my algorithm, compare the results to the existing baselines, and consider possible improvements.
7 (b)	I will deploy my algorithm in practice, for example, will put it on GitHub and encourage people to use it.
c	I will focus specifically on the implementation and application of the Naïve Bayes algorithm.
e	I will build a binary classification algorithm.
f	I will write some rules based on the keywords: for example, if an email contains a word “work”, it should be assigned with the “work-related” category, and so on.

1 (i)	I will analyse the task and consider how I would solve it myself.
2 (d)	I will inspect the data to see what it contains.
3 (j)	I will select features (e.g., words, word combinations, grammatical patterns) for the machine learning algorithm and extract them from the data.
4 (a)	I will consider using feature normalisation techniques.
5 (g)	I will implement and apply an appropriate classification algorithm capable of dealing with multiple classes.
6 (h)	I will thoroughly evaluate my algorithm, compare the results to the existing baselines, and consider possible improvements.
7 (b)	I will deploy my algorithm in practice, for example, will put it on GitHub and encourage people to use it.
c	I will focus specifically on the implementation and application of the Naïve Bayes algorithm.
e	I will build a binary classification algorithm.
f	I will write some rules based on the keywords: for example, if an email contains a word “work”, it should be assigned with the “work-related” category, and so on.

Introduction to text tokenization

Tokenization

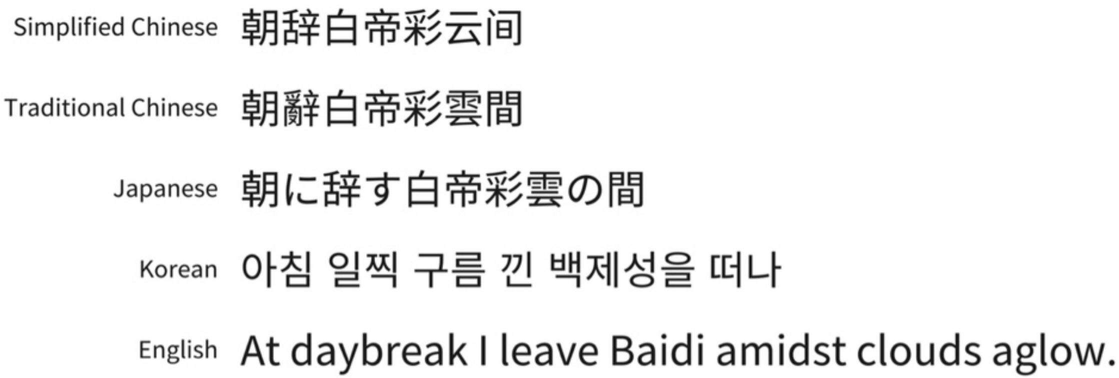
- For a machine, text comes in as a sequence of symbols, so it does not have an idea of what a **word** is
- **Tokenization** or **word segmentation** is the task of separating out (tokenizing) words in raw text
- First of all, how would you define a word?

Tokenization

- For a machine, text comes in as a sequence of symbols, so it does not have an idea of what a **word** is
- **Tokenization** or **word segmentation** is the task of separating out (tokenizing) words in raw text
- First of all, how would you define a word?
- The most straightforward solution – split by **whitespaces**
- Are there any problems with this solution? E.g., sequences of characters not containing whitespaces that should be split into multiple words, or single “words” containing whitespaces?

Tokenization

- It might be desirable to consider *New York*, *rock ‘n’ roll*, etc. single units (“words”): *New York* ≠ *New* + *York*
- Such sequences as *I’m*, *we’ve*, etc. should be split (*I am*, *we have*, etc.)
- Not all languages define words by whitespaces



Simplified Chinese	朝辞白帝彩云间
Traditional Chinese	朝辭白帝彩雲間
Japanese	朝に辞す白帝彩雲の間
Korean	아침 일찍 구름 낀 백제성을 떠나
English	At daybreak I leave Baidi amidst clouds aglow.

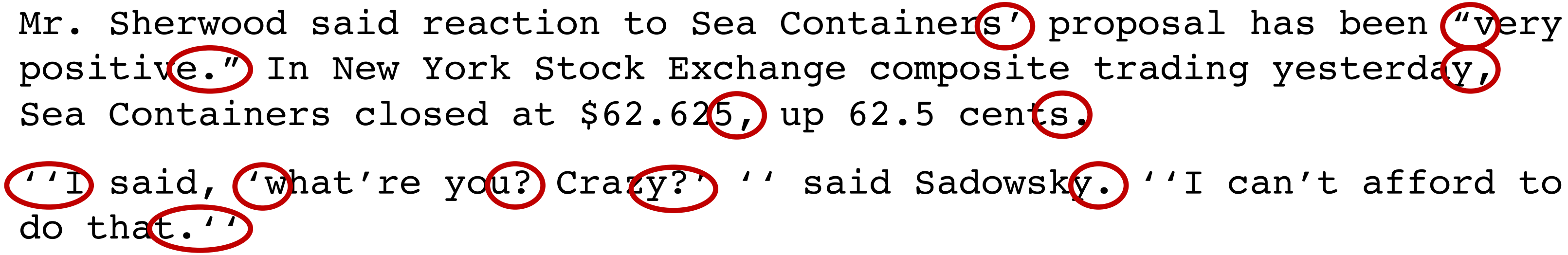
Tokenization step by step

Mr. Sherwood said reaction to Sea Containers’ proposal has been “very positive.” In New York Stock Exchange composite trading yesterday, Sea Containers closed at $62.625, up 62.5 cents.

‘‘I said, ‘what’re you? Crazy?’ ‘‘ said Sadowsky. ‘‘I can’t afford to do that.‘‘

- Let’s define patterns and use regular expressions to split this text into words
- What patterns can you define?
- Are there any challenges?

Tokenization step by step



Mr. Sherwood said reaction to Sea Containers’ proposal has been “very positive.” In New York Stock Exchange composite trading yesterday, Sea Containers closed at $62.625, up 62.5 cents.

‘‘I said, ‘what’re you? Crazy?’ ‘‘ said Sadowsky. ‘‘I can’t afford to do that.‘‘

```
re.split('\s+', s)
```

- Splitting by **whitespaces** will keep *Containers’*, *“very*, *positive.”*, etc. unsplit
- What else should be taken into account?

Tokenization step by step

Mr. Sherwood said reaction to Sea Containers’ proposal has been “very positive.” In New York Stock Exchange composite trading yesterday, Sea Containers closed at $62.625, up 62.5 cents.

‘‘I said, ‘what’re you? Crazy?’ ‘‘ said Sadowsky. ‘‘I can’t afford to do that.‘‘

```
re.split('([\s.,:;!?\'\”])+', s)
```

- Splitting by **whitespaces and punctuation marks** will solve the problem of, e.g., *Crazy?’* and *positive.”*
- **BUT** it will also split *Mr.* (as well as *Ph.D.*, *U.S.A.*, etc.) into multiple words
- This is undesirable. Can this be avoided? Are there any other problematic cases?

Tokenization step by step

Mr. Sherwood said reaction to Sea Containers’ proposal has been “very positive.” In New York Stock Exchange composite trading yesterday, Sea Containers closed at $62.625, up 62.5 cents.

‘‘I said, ‘what’re you? Crazy?’ ‘‘ said Sadowsky. ‘‘I can’t afford to do that.‘‘

- **Apostrophes** are ambiguous: *can’t* and *Containers’* should be split, but *cap’n* and *o’clock* shouldn’t
- **Number expressions** are challenging: *62.5*, *$62.625*, as well as *50,500* and *50 550,500* should not be split despite whitespaces and punctuation marks

Tokenization step by step

Mr. Sherwood said reaction to Sea Containers’ proposal has been “very positive.” In New York Stock Exchange composite trading yesterday, Sea Containers closed at $62.625, up 62.5 cents.

‘‘I said, ‘what’re you? Crazy?’ ‘‘ said Sadowsky. ‘‘I can’t afford to do that.‘‘

- Also, dates (*11/12/21* as well as *11.12.21*), URLs (*google.com*), email addresses (*google@gmail.com*), company names (*AT&T*) should be kept as single “words”
- Emoticons (*:)*) and hashtags (*#nlp*) should be kept as single “words”
- Expressions like *New York Times* should be identified as a single unit – these are called *multi-word expressions* and are dealt with using specialised algorithms

Tokenization in practice

- **Tokenization** is the first pre-processing step in many NLP applications: it has to be applied before other tools are applied, it has to run fast and be accurate
- In practice, you don’t need to invent your own tokenizer since all NLP (and ML) libraries and toolkits include highly optimized tokenizers
- These typically rely on the use of ML algorithms with a combination of regular expressions, lists of common abbreviations, and other techniques
- This week’s **homework**: implement your own simple tokenization algorithm based on regular expressions and compare it to one of the available ones from the **Natural Language Toolkit (NLTK)**

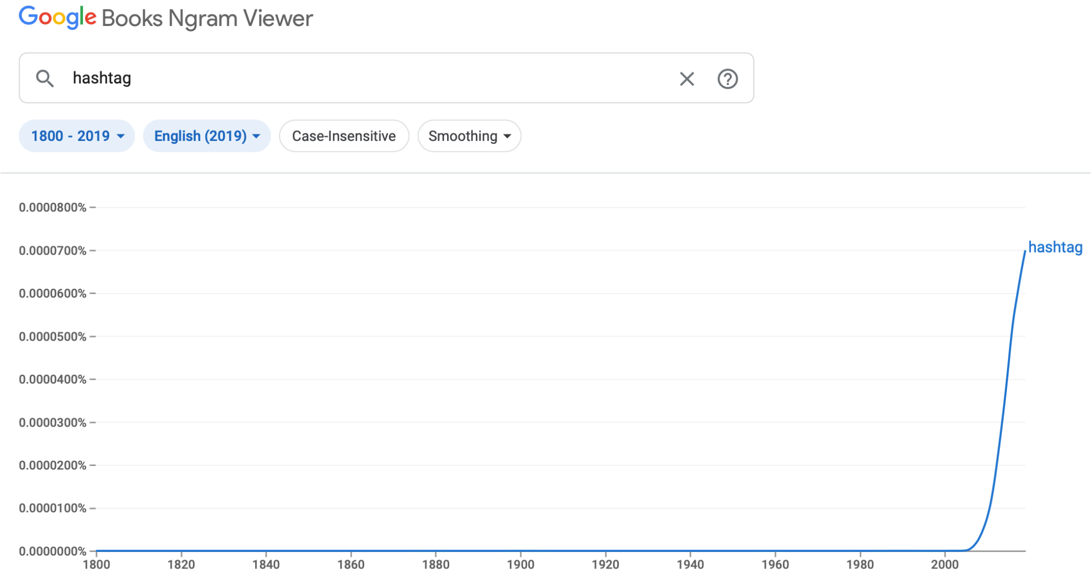
Introduction to linguistic analysis

Frequency analysis

- **Observation**: Words in language are not distributed evenly
- A few most frequent words used in language may cover the vast majority of all word occurrences in language: for example, 135 most frequent vocabulary items in English **account for 50%** of all words usage according to the Brown Corpus of American English
- Can you guess what the most frequent words in English are?

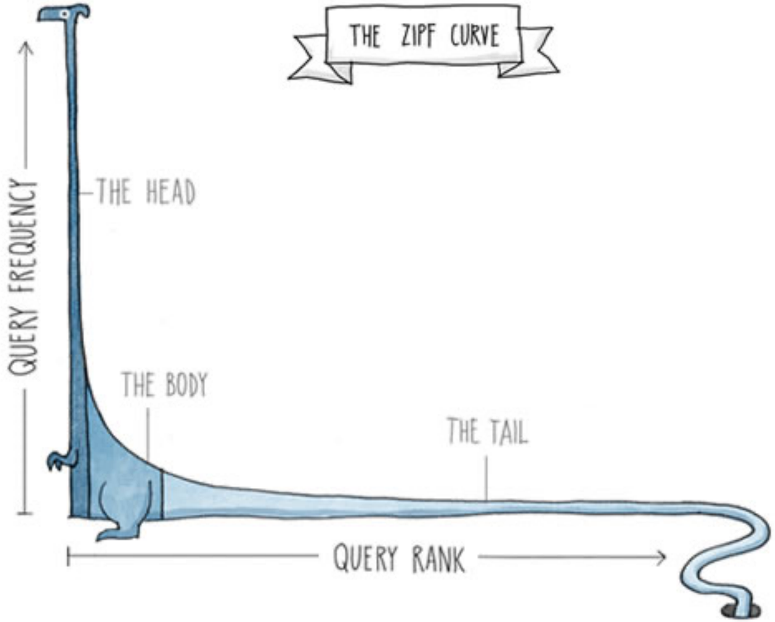
Frequency analysis

- **Observation**: Language is a dynamic system
- Words get added to it (invented or borrowed) all the time; other words may get out of fashion ⇒ hard to estimate at any particular point how many words a language actually contains



Word distribution

- **Observation**: given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table (the most frequent word is at rank 1)
- E.g.: in the Brown Corpus, the most frequent word *the* accounts for 7% of all word occurrences in this corpus (69,971 out of ~1mln), the second-ranked word *of* – around 3.5% (36,411 occurrences), followed by *and* (28,852), etc.



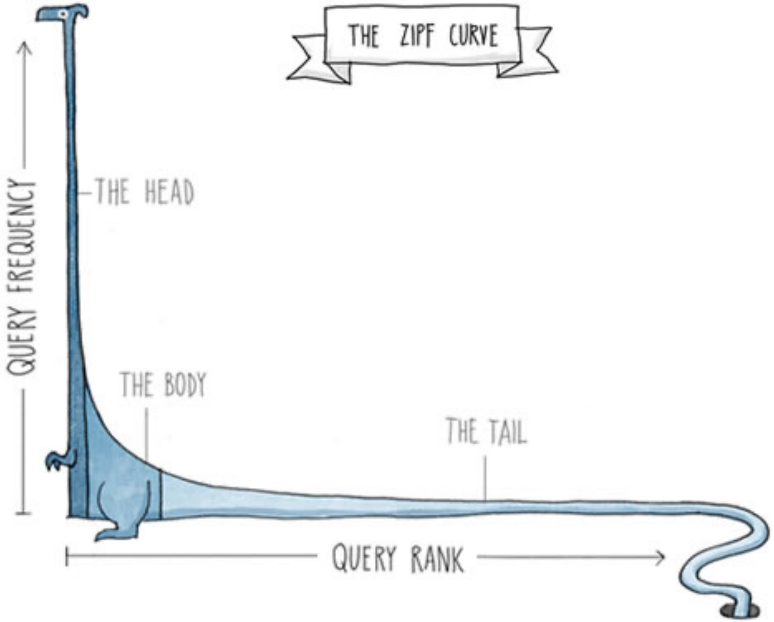
Zipf’s law

- After the American linguist George Kingsley Zipf
- In the population of *N* elements (e.g., *N* individual words in a corpus), the normalized frequency of the element of rank *k* (the fraction of the time the *k*-th most frequent word occurs) is defined as:

$$f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)}$$

  where *s* is the exponent: for English, *s*=1

- *s* is different for different languages
- (*s*=1.07 for the prediction of the cities’ population size based on cities’ size rank)



Implications

- If the most frequent words are *the*, *of*, *and*, *a*, *an*, *for*, *in*, *at*, *of*, *do*, *are*, and the like, and together they account for the vast majority of the word usage, what implications does it have for NLP tasks?

Implications

- If the most frequent words are *the*, *of*, *and*, *a*, *an*, *for*, *in*, *at*, *of*, *do*, *are*, and the like, and together they account for the vast majority of the word usage, what implications does it have for NLP tasks?
- Note that the words above help “glue” other, more meaningful words together, but by themselves they don’t express much meaning: e.g., *a book* vs *the book*, *stay in town* vs *stay out of town*
- In many contexts, such words are considered **stopwords** and in many applications they are filtered out (Steps 2-3 in the NLP pipeline)

Course logistics

Course objectives

- This course will help you acquire **theoretical knowledge** of the fundamental NLP concepts and techniques, as well as **practical skills**
- We will be looking into a **variety of NLP applications**
- Each concept will be explained from the perspective of its use in a particular application
- By the end of this course, you will be able to implement **your own NLP application** in an end-to-end manner

Course format

~20 lectures this semester (one 2-hour long lecture every Thursday)

Programming exercises will be provided every week after the lectures for you to get familiar with the techniques and implementation. Exercises are left as your homework and are not assessed. Solutions will be made available on Mondays

Lab sessions are optional: you can use this time to work on the practical exercises especially if using University resources is preferable

Assessment

- 30% for coursework (mini-project)
- 70% for the exam

Coursework (mini-project)

- Your task is to build a **sentiment analysis application** that can automatically detect whether a movie review is positive or negative
- Reviews are extracted from the IMDb database. The dataset for this project and the task description will be released tomorrow, **Friday, October 7**
- You will be required to **submit a report** detailing your implementation steps: the report should be included in your Jupyter notebook together with the accompanying code (if you want to split your implementation into multiple notebooks, you can submit a .zip file, but one of the notebooks has to contain the main report)

Coursework (mini-project)

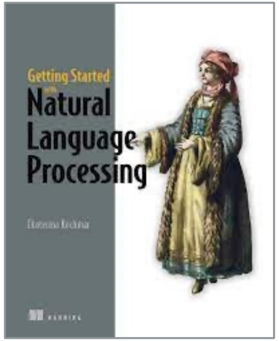
- Your report should detail all the steps of the NLP pipeline and explain the decisions that you've made
- You will be assessed on the basis of your report and not just on the basis of the results achieved by of your algorithm
- **Submission deadline**: Monday, December 12, 8pm

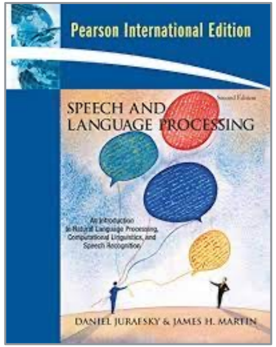
Course resources

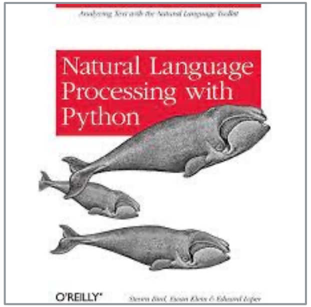
- Lecture slides and recordings
- Handouts will be released after lectures
- You are expected to work individually: e.g., handouts have suggested activities. Such activities are not obligatory and are not assessed, however, you are encouraged to attempt the tasks and exchange your observations on the Moodle forum
- Programming exercises are your homework – they are not assessed
- Problem sets (tasks of the type you may get in the exam) – will be released during the revision week

Books

- Kochmar, E. (2021). *Getting Started with Natural Language Processing* – available online via the University of Bath Library
- Jurafsky, D. and Martin, J.M. (2009). *Speech and Language Processing* – available at https://github.com/rain1024/slp2-pdf/tree/master/chapter-wise-pdf and https://web.stanford.edu/~jurafsky/slp3/
- Bird, S., Klein, E., and Loper E. (2009). *Natural Language Processing with Python* – available at http://www.nltk.org/book/




Pearson International Edition
Second Edition
SPEECH AND LANGUAGE PROCESSING
An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition
DANIEL JURAFSKY & JAMES H. MARTIN


Analyzing Text with the Natural Language Toolkit
Natural Language Processing with Python
O'REILLY
Steven Bird, Ewan Klein & Edward Loper

NLP libraries and toolkits

- NLTK
- SpaCy
- Gensim
- Matplotlib
- NumPy
- Scikit-learn
- Installation instructions available on Moodle
- You can either install the libraries on your own computer or run the code in Google's Colab