Applied Deep Learning for NLP

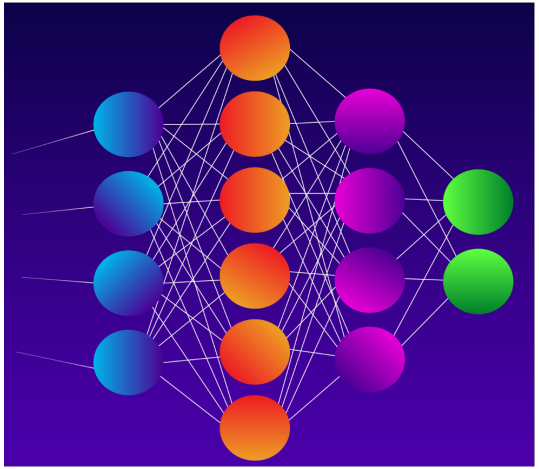
Week 1 - NLP Intro

Juan Carlos Medina Serrano

Technische Universität München
Hochschule für Politik
Political Data Science

Munich, 22. October 2020

political
data
science



Seminar Details

Weekly Meeting: Thursdays **11:30-13:00**

- **Coding** + **Lecture** on NLP and Deep Learning
- Coding notebook will sometimes extra codes at the end. This is for you to check out after the course is over.
- **Final Project** Develop an Alexa skill with NLP. AWS usage is optional
- Weekly tasks to advance the final project.
  Groups of 2 Pairing will be done next week.
- Use TUM chat groups to stay active in the course! Contact me with a direct message instead of an email, please
- No final exam. So important to attend meetings. Otherwise, I know learning by yourself = low priority
- Check the Wiki for more seminar information.

Resources

Books:

- **Practical Natural Language Processing** Vajjala, Majumder, Gupta, Surana
- **Natural Language Processing in Action** Lane, Howard, Hapke
- **Applied Text Analysis with Python** Bengfort, Bilbro, Ojeda
- **Deep Learning for Natural Language Processing** Goyal, Pandey, Jain
- **Hands on Machine Learning with Scikit-learn and Tensorflow 2.0** Geron. *A must read for machine learners*

Online courses:

- **Introduction to Deep Learning** TUM `https://www.youtube.com/watch?v=QLOocPbztuc&list=PLQ8Y4kIIbzy_OaXv86lfbQwPHSomk2o2e&ab_channel=MatthiasNiessner`
- **Natural Language Processing with Deep Learning** Stanford `https://www.youtube.com/watch?v=8rXD5-xhemo&list=PLoROMvodv4rOhcuXMZkNm7j3fVwBBY42z&ab_channel=stanfordonline`
- **A Code-First Introduction to Natural Language Processing** Fast AI `https://www.fast.ai/2019/07/08/fastai-nlp/`
- **Deep Learning** NYU `https://atcold.github.io/pytorch-Deep-Learning/`. `https://www.youtube.com/watch?v=0bMe_vCZo30&list=PLLHTzKZzVU9eaEyErdV26ikyolxOsz6mq&ab_channel=AlfredoCanziani`

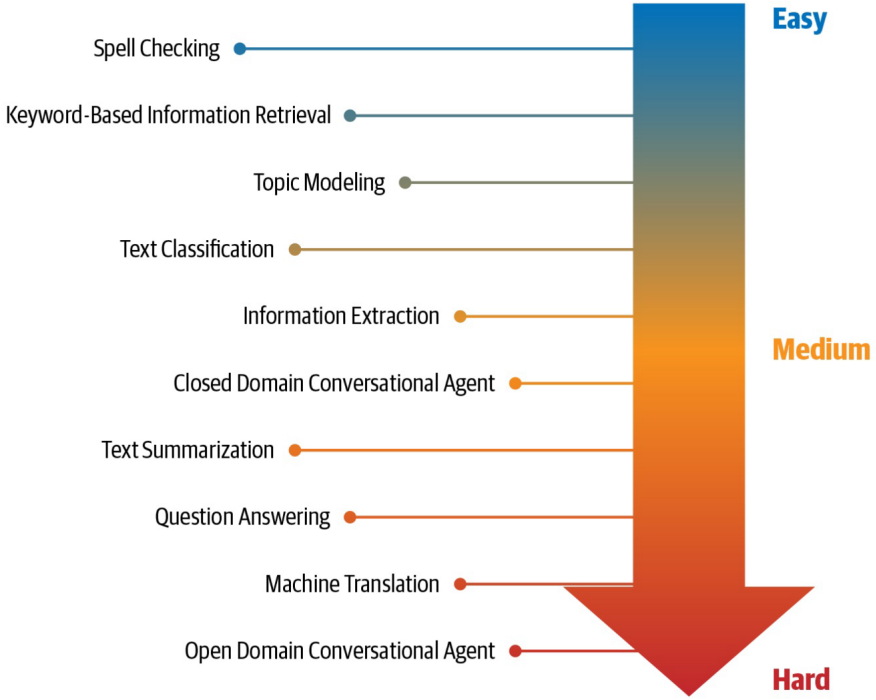
NLP Applications

Search	Web	Documents	Autocomplete
Editing	Spelling	Grammar	Style
Dialog	Chatbot	Assistant	Scheduling
Writing	Index	Concordance	Table of contents
Email	Spam filter	Classification	Prioritization
Text mining	Summarization	Knowledge extraction	Medical diagnoses
Law	Legal inference	Precedent search	Subpoena classification
News	Event detection	Fact checking	Headline composition
Attribution	Plagiarism detection	Literary forensics	Style coaching
Sentiment analysis	Community morale monitoring	Product review triage	Customer care
Behavior prediction	Finance	Election forecasting	Marketing
Creative writing	Movie scripts	Poetry	Song lyrics

NLP Tasks

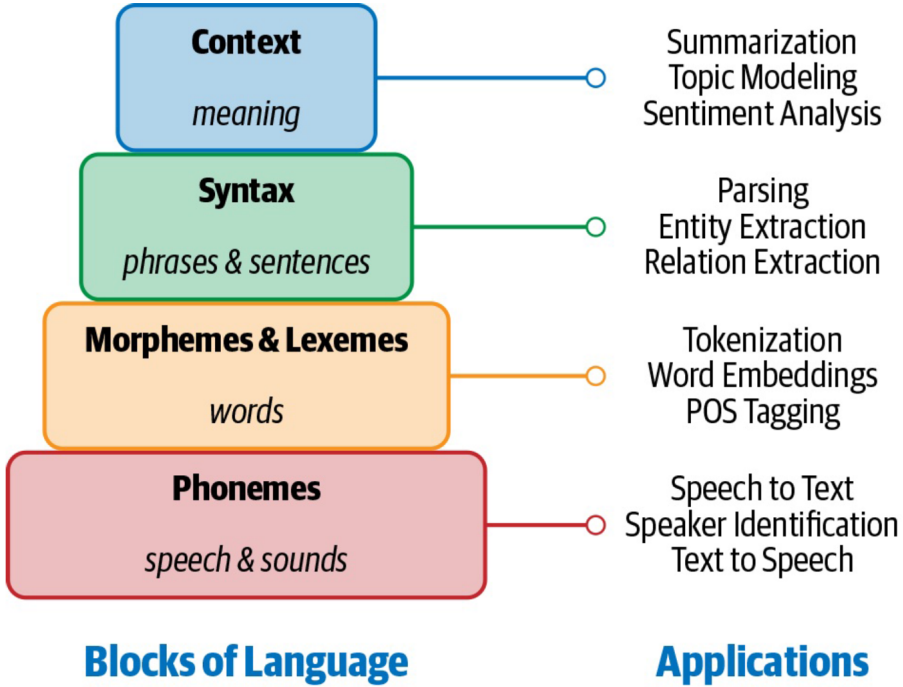
- **Language Modeling** Predicting what the next word in a sentence will be based on the history of previous words. Learn the probability of sequence of words.
- **Text Classification** Classifying into a set of categories based on content (email spam)
- **Information Extraction** Extracting relevant information from text
- **Information Retrieval** Finding documents relevant to a user query
- **Conversational Agent** Building a dialogue system to converse
- **Text Summarization** Create short summaries of longer documments
- **Question Answering**
- **Machine Translation**
- **Topic Modeling** Uncovers the topical structure of a large collection of documents.

NLP Tasks


Easy
Spell Checking
Keyword-Based Information Retrieval
Topic Modeling
Text Classification
Information Extraction
Medium
Closed Domain Conversational Agent
Text Summarization
Question Answering
Machine Translation
Open Domain Conversational Agent
Hard

Language Blocks

LINGUISTICS The study of language



Phonemes

Smallest units of sound.

Consonant phonemes, with sample words		Vowel phonemes, with sample words	
1. /b/ - bat	13. /s/ - sun	1. /a/ - ant	13. /oi/ - coin
2. /k/ - cat	14. /t/ - tap	2. /e/ - egg	14. /ar/ - farm
3. /d/ - dog	15. /v/ - van	3. /i/ - in	15. /or/ - for
4. /f/ - fan	16. /w/ - wig	4. /o/ - on	16. /ur/ - hurt
5. /g/ - go	17. /y/ - yes	5. /u/ - up	17. /air/ - fair
6. /h/ - hen	18. /z/ - zip	6. /ai/ - rain	18. /ear/ - dear
7. /j/ - jet	19. /sh/ - shop	7. /ee/ - feet	19. /ure/[4] - sure
8. /l/ - leg	20. /ch/ - chip	8. /igh/ - night	20. /ə/ - corner (the 'schwa' - an unstressed vowel sound which is close to /u/)
9. /m/ - map	21. /th/ - thin	9. /oa/ - boat	
10. /n/ - net	22. /**th**/ - then	10. /**oo**/ - boot	
11. /p/ - pen	23. /ng/ - ring	11. /oo/ - look	
12. /r/ - rat	24. /zh/[3] - vision	12. /ow/ - cow	

Morphemes and Lexemes

MORPHEMES: Smallest unit of language that has a meaning. Not all morphemes are words (prefixes, suffixes)

unbreakable
un + break + able

cats
cat + s

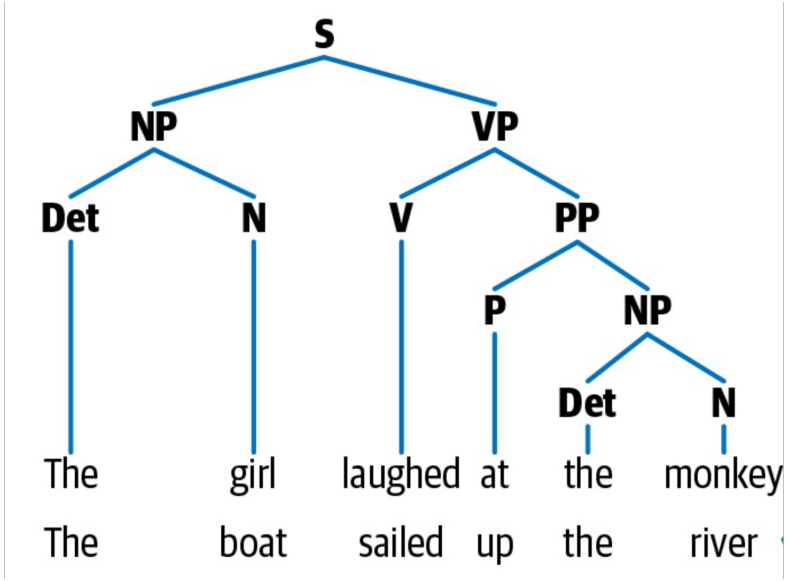
tumbling
tumble + ing

unreliability
un + rely + able + ity

LEXEMES: Structural variations of morphemes related to one another by meaning. *run* and *running* belong to the same lexeme form

Syntax

Rules to construct grammatically correct sentences in a language.
Common to represent sentences: **parse tree**



Words at the lowest level, followed by **part-of-speech** (POS) tags, followed by **phrases** (NP:Noun phrase, VP:Verb phrase,), and ending with **sentence** at the highest level.

Syntax

** Missing: **Clauses** exist between phrases and sentences and are joined by conjuctions.

Example: The girl laughed at the monkey **and** the boat sailed up the river.

Each POS tag like the noun (N) can be further subdivided into categories like singular nouns (NN), singular proper nouns (NNP), and plural nouns (NNS)

A complete list of POS tags:
`https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`

Context

How various parts of language come together to convey a particular meaning. Composed of:

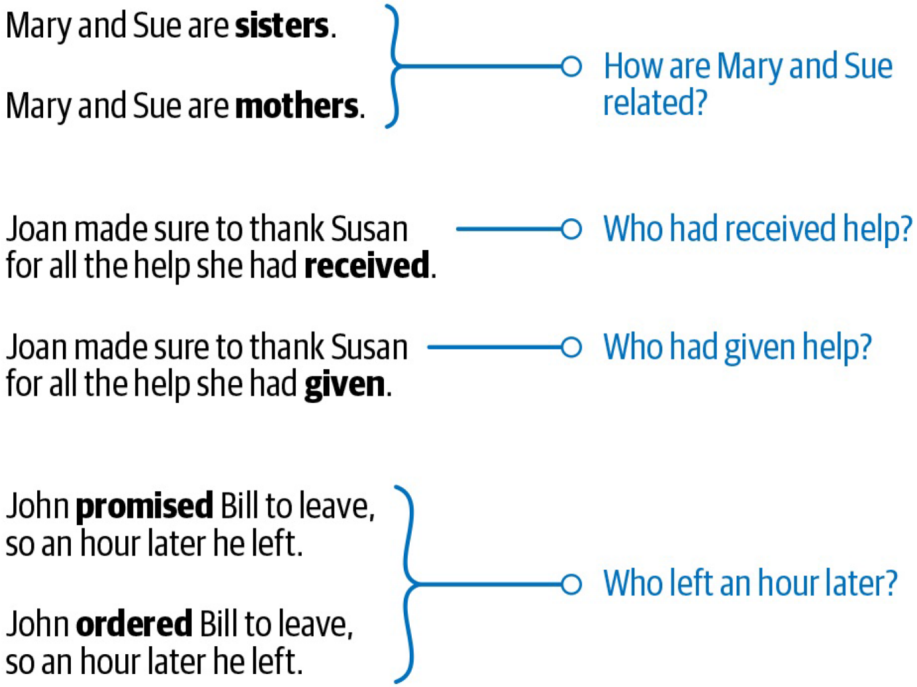
- *Semantics* Direct meaning of words without external contents
- *Pragmatics* Adds word knowledge and external context.

Why is NLP hard?

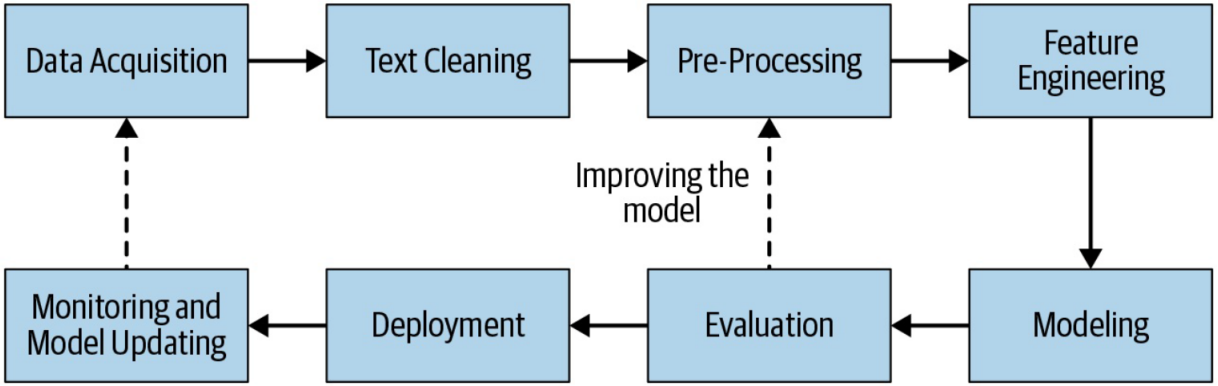
- Ambiguity
- Common knowledge
- Creativity
- Diversity across languages

Why is NLP hard?

AMBIGUITY:



NLP Pipeline

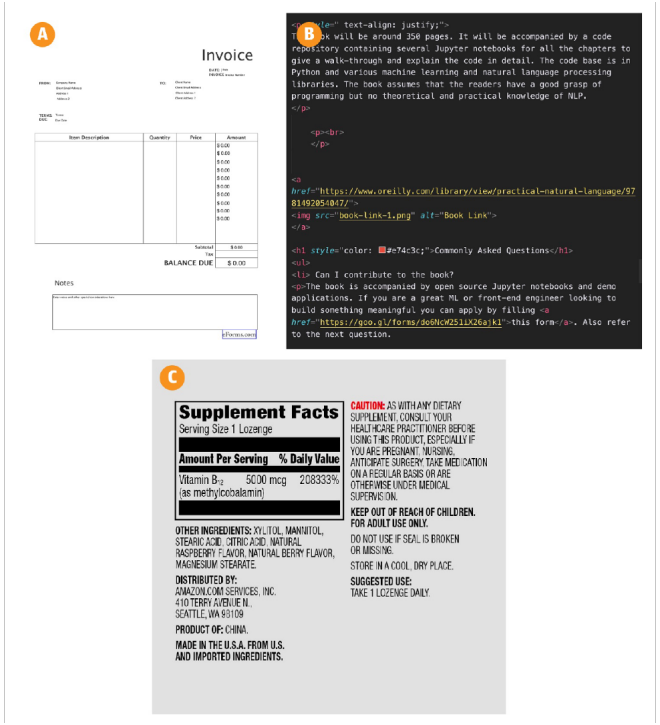


Data Acquisition

- Use a public dataset. Example: `https://datasetsearch.research.google.com/`
- Use a private dataset. (Someone gave it to you, example, a company)
- Scrape data!
- Data augmentation
  - Synonym replacement
  - Back translation
  - Replacing entities
  - Adding noise. (Spelling mistakes, words written similar)

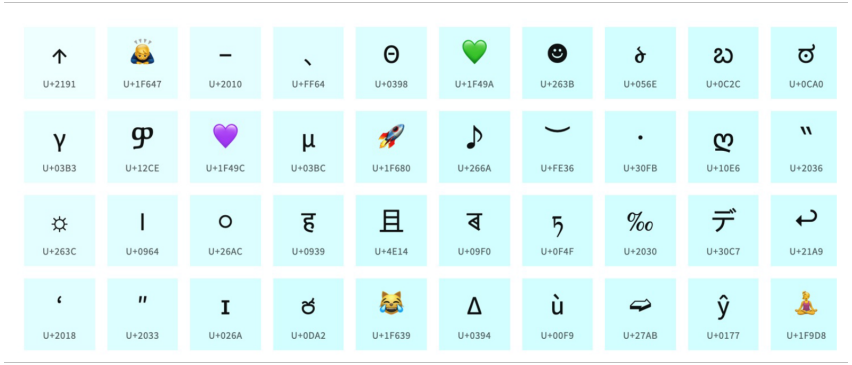
Text Cleaning

Extracting raw text from input data and removing all the other non-textual information, and convert the text to a required encoding format.



Text Cleaning

- **Unicode normalization**



- **Spelling correction**
  - Shorthand typing: *Hllo world!*
  - Fat finger problem: *I pronise her to do the homeworl*

Pre-processing

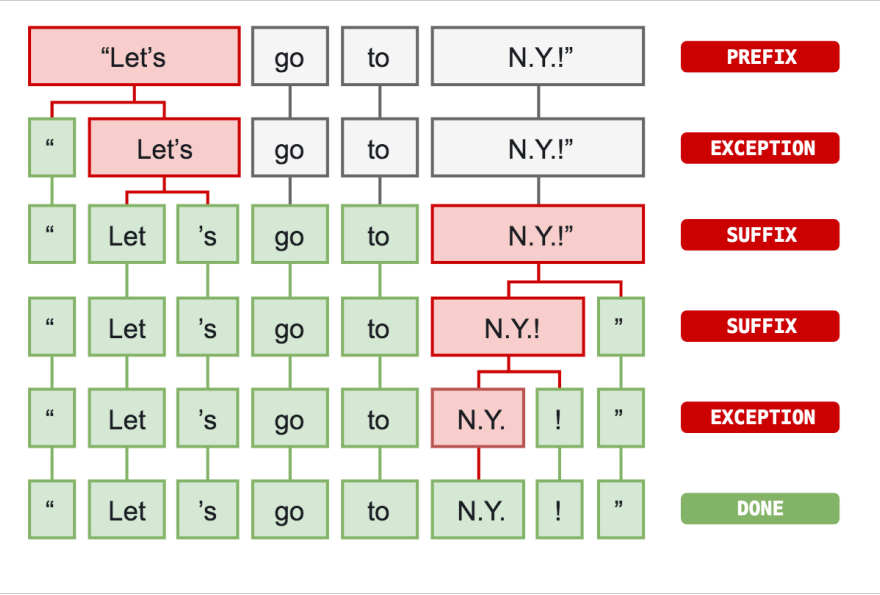
- **Preliminaries:** Sentence segmentation and word tokenization
- **Frequent steps:** Stop word removal, stemming, lemmatization, digits/punctuation removal, lowercasing
- **Advanced:** POS tagging, parsing, coreference resolution.
- **Others:** Normalization, language detection ...

The order of the pre-processing steps is important! Think always what you want the text input to the model to be like.

Sentence Segmentation and Word Tokenization

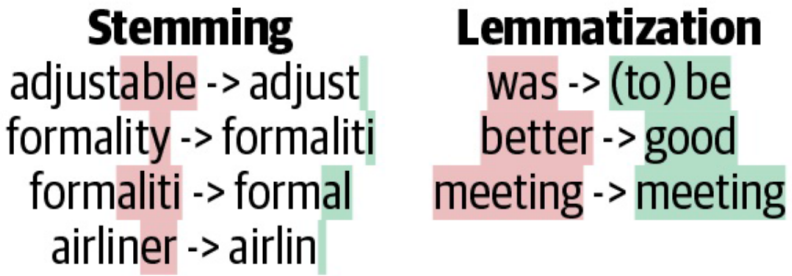
Spliting the text into sentences and further split a sentence into words.

Every NLP library has different tokenizers. Some of them are special: for example, NLTK's tweet tokenizer. Does not separate hashtag from word.
They are language-specific!



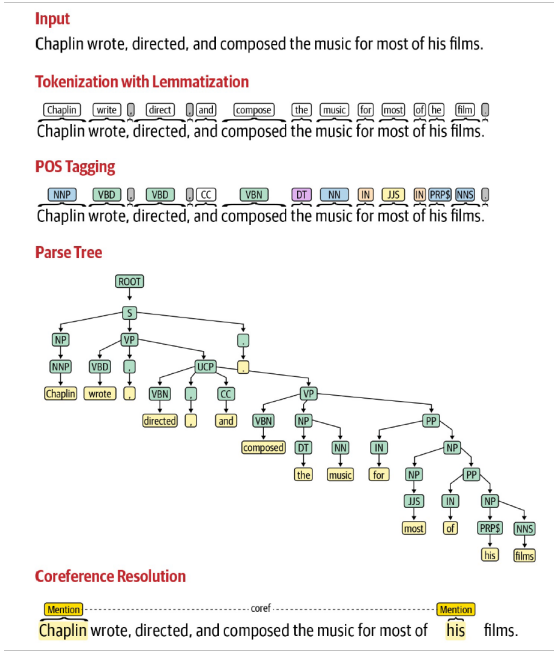
Frequent Steps

- Removing **stop words** Frequent words in a language that (most of the time) do not carry any useful information. (a, an, the, of, in,...)
- Removing **punctuation** and *digits*
- **Lowercasing**
- **Stemming** Removing suffixes and reducing a word to a base form. (Used in search engines or to reduce feature space)
- **Lemmatization** Mapping the different forms of a word to its base word (*lemma*)



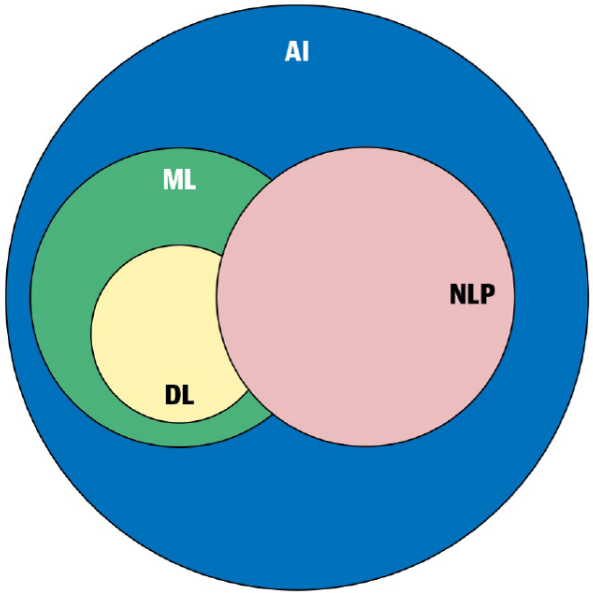
Advanced Pre-processing

- **POS tagging**
- **Coreference resolution** Identify patterns indicating a relation between two entities in a sentence. It uses a parse tree for this.



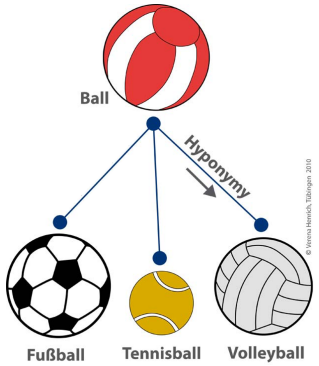
NLP Paradigms

Before moving on with the NLP Pipeline, we need to understand the different approaches to NLP



Heuristic-Based Approaches

- Using **dictionaries** and thesauruses.
  Example: Sentiment Analysis dictionaries. Count number of positive and negative words in the text.
- **Wordnet** Database of words with relationships. Includes **synonyms** and **hyponyms** (capture is-type-of relationships).

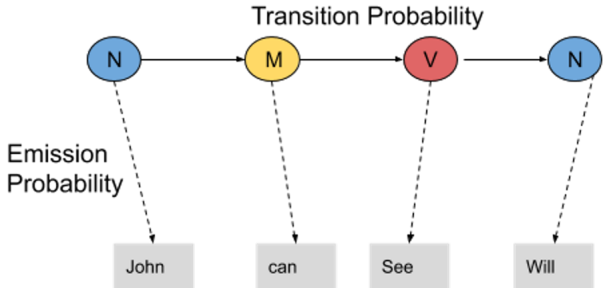


- **Regular Expressions** A pattern that is used to match and find substrings in a text.
  Example: Finding all emails in a text:
  $\wedge([a-zA-Z0-9\_\-\.]+)@([a-zA-Z0-9\-\.]+)\.([azA-Z]\{2,5\})\$$
- **Context-free grammars** Check additional slides on Moodle!

Machine Learning

Supervised and unsupervised machine learning algorithms. Popular in NLP:

- **Naive Bayes**
- **Logistic Regression**
- **Support Vector Machines**
- **Hidden Markov Model** Assumes there is an unobservable process with hidden states from the data. Each hidden state depends on previous steps. Great for POS tagging!

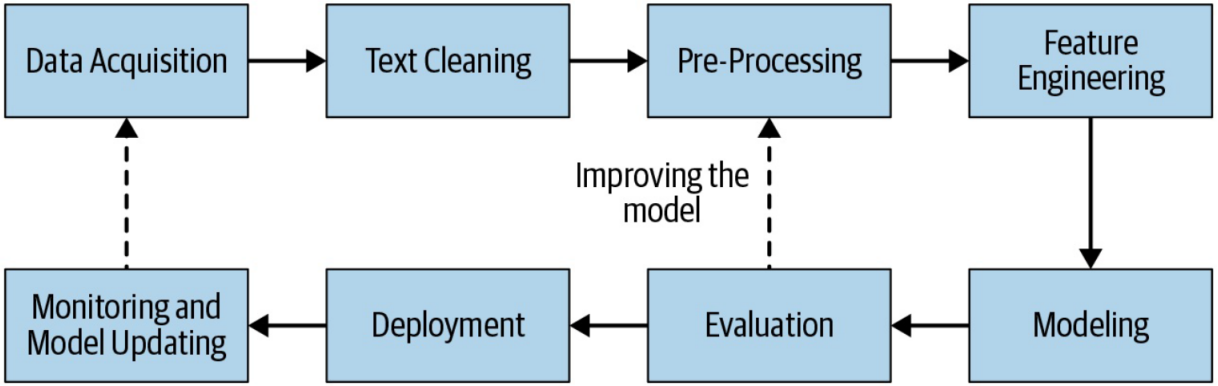


- **Latent Dirichlet Allocation** The famous topic modeling algorithm

Deep Learning

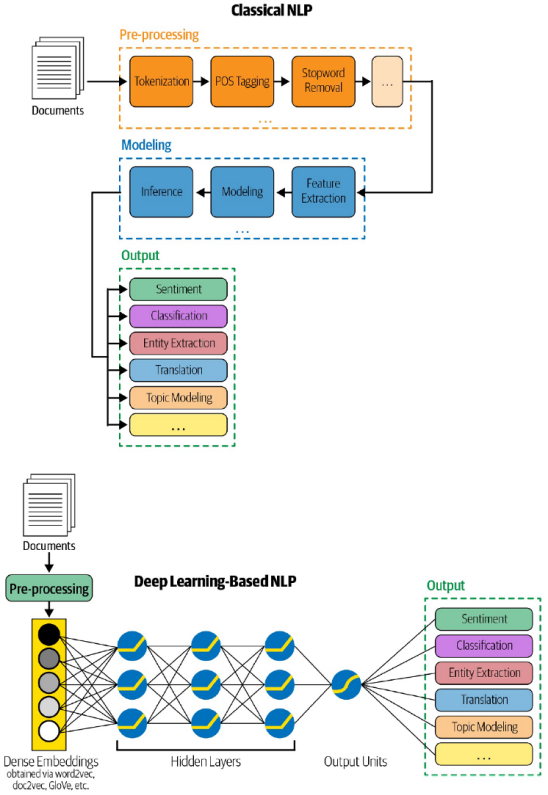
What this seminar is about! Starting next week!

NLP Pipeline



Feature Engineering

Capture the characteristics of text into numeric vectors that can be understood by the algorithms

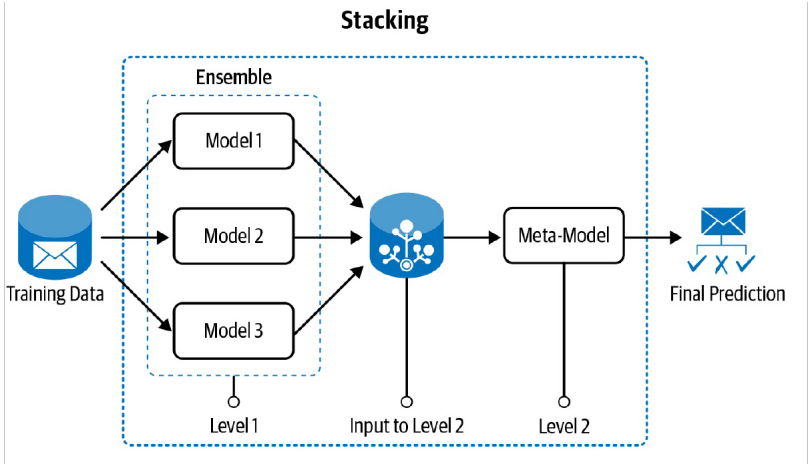


Modeling

1. Start with simple heuristics.
2. Build off-the-shelf models (even from APIs). Helpful as baselines. Use the heuristics from step 1, either:
   - As features - when the behavior of combined features is fuzzy
   - As pre-precessing step - highly deterministic features.
3. Build your custom model:

Modeling

ENSEMBLES Pool predictions from multiple models.
STACKING Feed one model's output as input for another model.



TRANSFER LEARNING Transfers preexisting knowledge from a big, well-trained model to a newer model at its initial phase (BERT) New in NLP since 2018!
Reapplying heuristics at the end as control for model errors or **Human-in-the-loop** controls

Evaluation

Measure how good the model is! Two types of **metrics**:

- **Intrinsic** These are the ones typical in machine learning. Compares some **ground truth** to the output of the model.
  Important metrics: Accuracy, Precision, Recall, F1 score, AUC, MRR, MAP, RMSE, BLEU, METEOR, Perplexity

- **Extrinsic** Fous on the final objective. ´´Real world"(business) metrics. For example, does it solve a business problem? Does it bring more customers?

Normally, intrinsic metrics evaluated before extrinsic metrics. Bad intrinsic most of the times leads to bad extrinsic. However, the contrary may not be true.

Final Steps

DEPLOYMENT Deploy in a production environment. Can be deployed as a web service either on the cloud or on your own server.

We will use AWS and create an Alexa skill as a product.

MONITORING

MODEL UPDATING Online learning model, manual labeling on results, make model faster (caching, bigger computing power), make model slower (use less resources as expected)

Pipeline Example

Uber's Customer Care System