



BERN WINTER SCHOOL – NATURAL LANGUAGE PROCESSING

Ahmad Alhineidi

Day 2 – Content

- 08:15 – 09:00 : Introduction to NLP tasks
- 09:00 – 10:15: Different methods in NLP
- 10:15 – 10:45: Break
- 10:45 – 12:30: Word embeddings
- 12:30 – 17:00: Break
- 17:00 – 18:30: Open-source NLP in Python

Materials on Github

- https://github.com/dsl-unibe-ch/Winter_School_NLP

Natural Language?

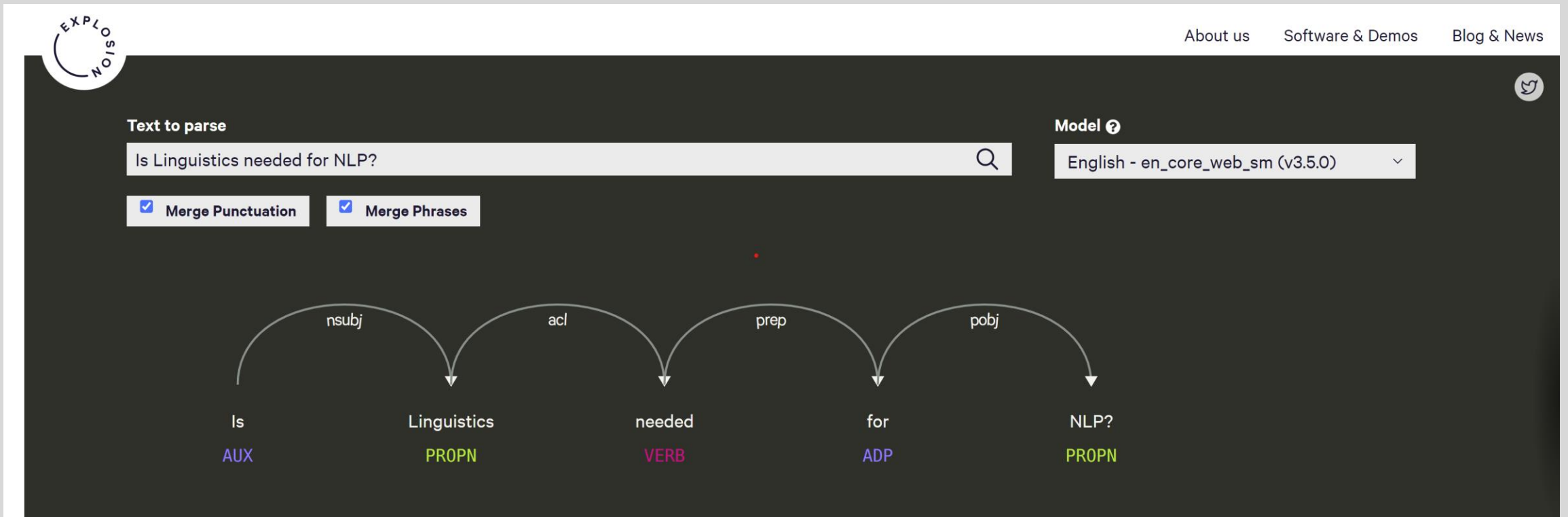
- “...network of constructions..”
- -“C is a construction iff C is a form- meaning pair $\langle F, S \rangle$ such that some aspects of F or some aspects of S is not strictly predictable from C’s component parts or from other previously established constructions.”

Goldberg, A. E. (1995). *Constructions: A construction grammar approach to argument structure*. University of Chicago Press.

[Adele Goldberg on Linguistics and Grammar \(Youtube\)](#)

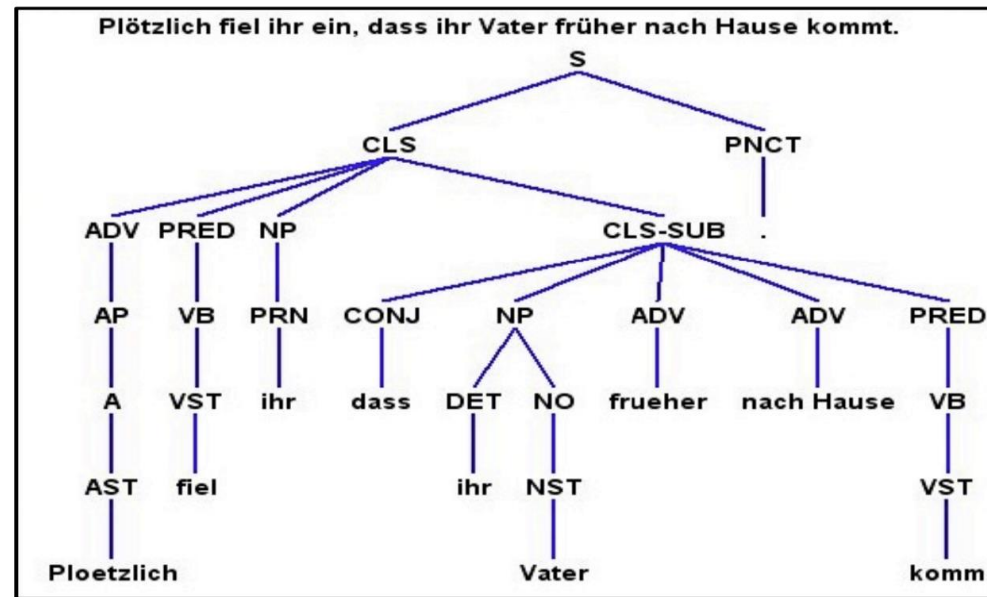
Linguistics for NLP?

- How much linguistic knowledge needed for NLP?
- Will a POS tagged corpus perform better as training data for machine learning algorithm?



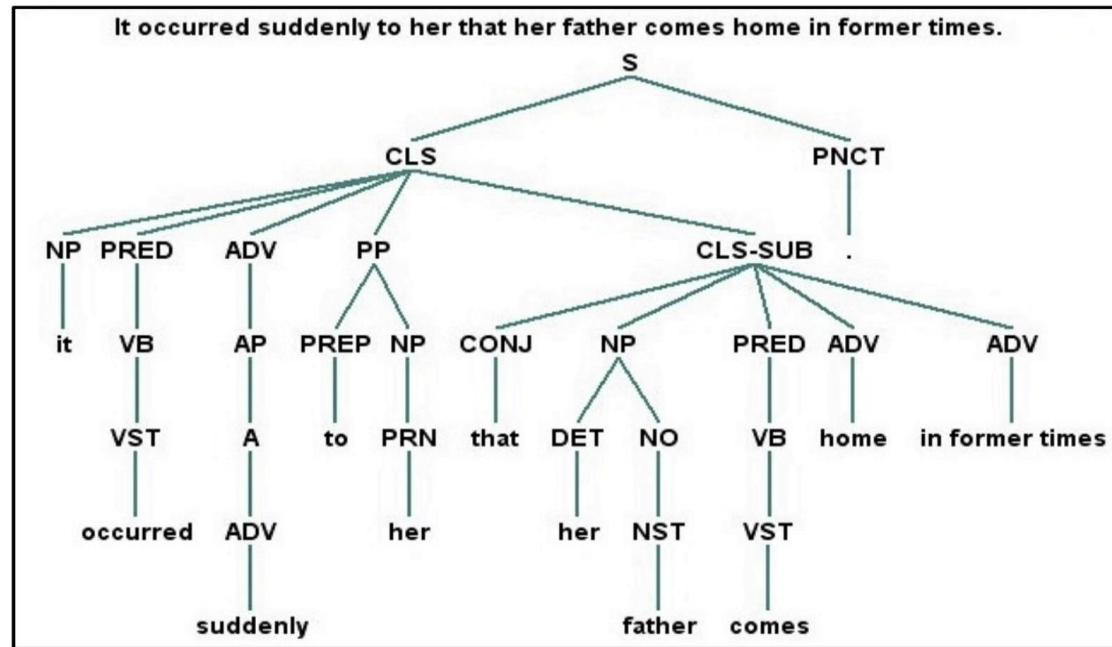
Linguistics for NLP?

Old machine translation system relied on linguistic knowledge



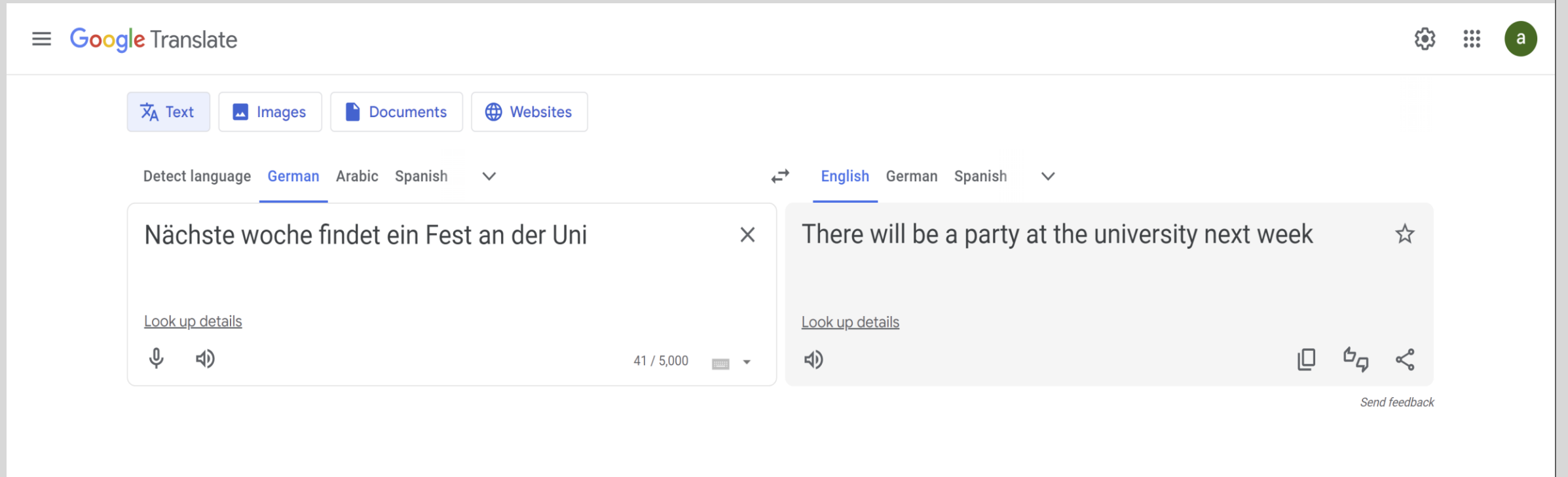
Linguistics for NLP?

Old machine translation system relied on linguistic knowledge



Linguistics for NLP?

New machine translation system predict the next word or sequence



The screenshot displays the Google Translate interface. At the top, the Google Translate logo is on the left, and settings, app drawer, and user profile icons are on the right. Below the header, there are four tabs: 'Text', 'Images', 'Documents', and 'Websites'. The 'Text' tab is selected. The interface is split into two main sections. The left section, labeled 'Detect language', shows 'German' as the source language. It contains a text input box with the German sentence 'Nächste woche findet ein Fest an der Uni'. Below the input box are links for 'Look up details', a microphone icon, a speaker icon, and a character count '41 / 5,000'. The right section, labeled 'English', shows the translated sentence 'There will be a party at the university next week'. It also includes a 'Look up details' link, a speaker icon, and icons for copying, sharing, and feedback. A star icon is visible in the top right corner of the right section.

Google Translate

Text Images Documents Websites

Detect language **German** Arabic Spanish

Nächste woche findet ein Fest an der Uni

[Look up details](#)

41 / 5,000

English German Spanish

There will be a party at the university next week

[Look up details](#)

Send feedback

Linguistics for NLP?


- “Anytime a linguist leaves the group, the recognition rate goes up”

(1988), Fred Jelinek, pioneer in Statistical
methods for
Speech Recognition

NLP tasks

- Source: Innerdoc [link](#)

Periodic Table of Natural Language Processing Tasks

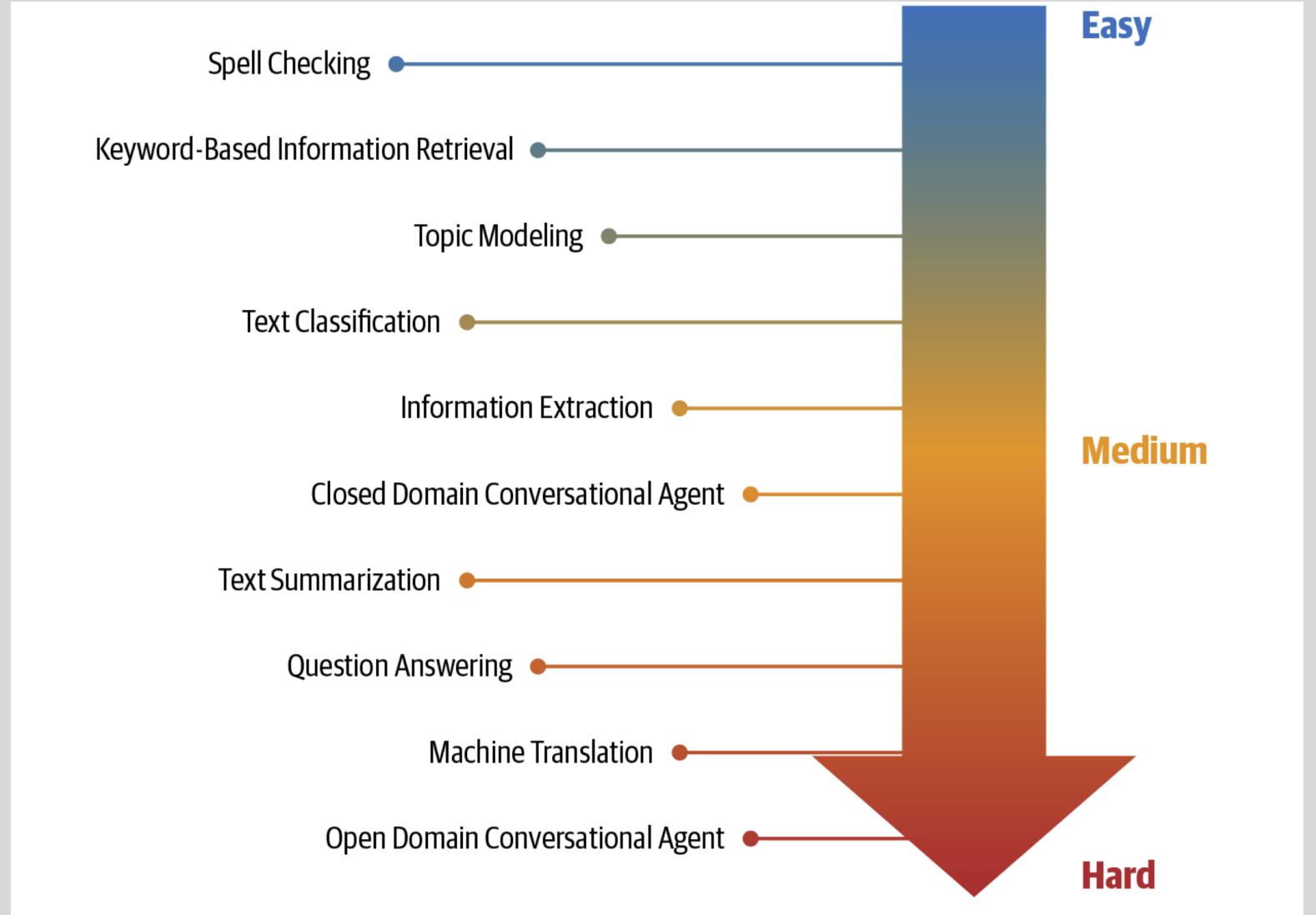
1 Bit Bits to Character Encoding															 www.innerdoc.com														75 App Interactive App Creation				
2 Typ Manual Typewriting	8 Man Manual Annotation															29 Pri Price Parser															63 Nex Next Token Prediction	69 Rel Relation Extraction	76 Ann Annotated Text Visualization
3 Str Loading a Structured Datafile	9 Act Annotation with Active Learning	14 Tok Tokenization	19 Ste Stemming	24 Ngr N-grams	30 Geo Geocoding															43 Trn Training Models	48 Spa Spam Detection	53 Key Keyword Extraction	58 Syn Wordnet Synsets	64 Rep Report Writing	70 Qan Question Answering	77 Wcl Wordcloud							
4 Cor Generating a Corpus	10 Pro Training Data Provider	15 Voc Vocabulary Building	20 Lem Lemmatization	25 Phr Rulebased Phrasematcher	31 Tmp Temporal Parser	35 Sen Sentencizer	39 Ded Deduplication	44 Tst Evaluating Models	49 Sed Sentiment and Emotion Detection	54 Esu Extractive Summarization	59 Dst Distance Measures	65 Tra Machine Translation	71 Cha Chatbot Dialogue	78 Emb Word Embedding Visualization																			
5 Api Loading from API	11 Cro Crowdsourcing Marketplace	16 Mor Morphological Tagger	21 Nrm Normalization	26 Chu Dependency Nounchunks	32 Nel Named Entity Linking	36 Par Paragraph Segmentation	40 Raw Raw Tekst Cleaning	45 Exp Explaining Models	50 Int Intent Classification	55 Top Topic Modeling	60 Sim Document Similarity	66 Asu Abstractive Summarization	72 Sem Semantic Search Indexing	79 Tim Events on Timeline																			
6 Scr Text and File Scraping	12 Aug Textual Data Augmentation	17 Pos Part-of-Speech Tagger	22 Spl Spell Checker	27 Ner Named Entity Recognition	33 Crf Coreference Resolution	37 Grm Grammar Checker	41 Met Meta-Info Extractor	46 Dpl Deploying Models	51 Cls Text Classification	56 Tre Trend Detection	61 Dis Distributed Word Representations	67 Prp Paraphrasing	73 Kno Knowledge Base Population	80 Map Locations on Geomap																			
7 Ext Text Extraction and OCR	13 Rul Rulebased Training Data	18 Dep Dependency Parser	23 Neg Negation Recognizer	28 Abr Abbreviation Finder	34 Anm Text Anonymizer	38 Rea Readability Scoring	42 Lng Language Identification	47 Mon Monitoring Models	52 Mlc Multi-Label Multi-Class Classification	57 Out Outlier Detection	62 Con Contextualized Word Representations	68 Lon Long Text Generation	74 Edi E-Discovery and Media Monitoring	81 Gra Knowledge Graph Visualization																			
Source Data Loading	Training Data Generation	Word Parsing	Word Processing	Phrases and Entities	Entity Enriching	Sentences and Paragraphs	Documents	Model Development	Supervised Classification	Unsupervised Signaling	Similarity	Natural Language Generation	Systems	Information Visualization																			

Common NLP tasks and applications

- Text classification (Sentiment analysis, spam detection, topic labeling)
- Named Entity Recognition (NER) (Information extraction, content recommendation)
- Machine Translation (Content Localization, real-time translation)
- Text Summarization (News Aggregation, Research)
- Question Answering (Customer Support)
- Speech Recognition (Voice Assistants)
- Text Generation (Content Creation, chatbots)



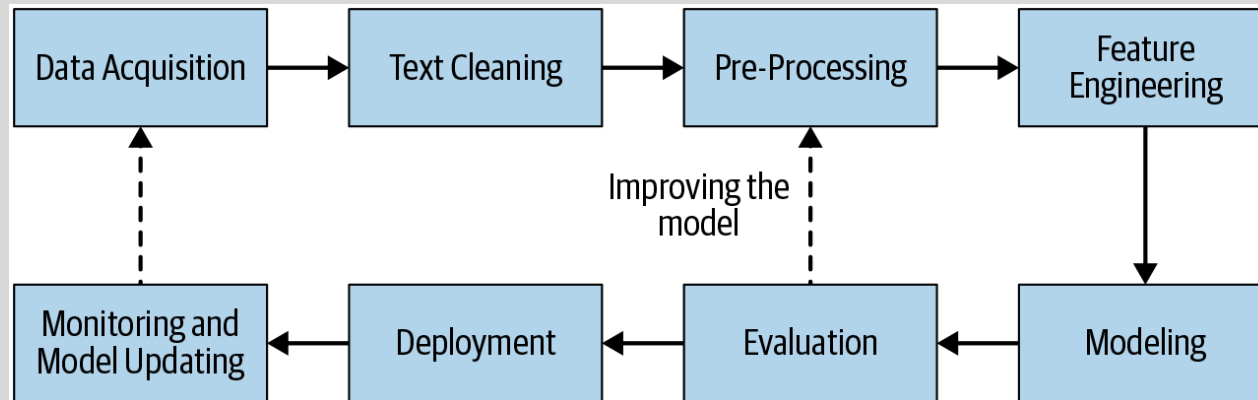
Common NLP tasks



Source: Vajjala et al. 2020

NLP process

Generic NLP Pipeline



Source: Vajjala et al. 2020

NLP progress?



The model with the best performance wins. Is that the only metric?



Check
<https://nlppprogress.com/>

Commercial NLP APIs

- Explore one the demos of commercial tools for NLP (text mining):

1. [Watson NLU](#)
2. [Google Cloud Natural Language](#)
3. [Klangoo magnet](#)
4. [Pikes](#)
5. [TextRazor](#)
6. [text2data](#)
7. [Dandelion](#)
8. [Gate Cloud](#)

- Give a feedback of what you like and dislike (5-10min)
- Use the link to add screenshot, comments of your findings [link](#)

NLP approaches



RULE AND LOGICAL
BASED



STATISTICAL
(CLASSICAL ML) MODELS



NEURAL NETWORKS
(DEEP LEARNING)

NLP approaches

- **Source:** Schmidt, Thomas, et al. "Sentiment analysis on twitter for the major German parties during the 2021 German federal election." *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)*. 2022.

	SVM	NB	GerVADER	BERT-1	BERT-2	BERT-3
Accuracy	57.6	65.0	52.0	85.8	81.5	93.3
F1 Macro	54.5	65.3	52.0	82.1	73.8	93.4
F1 Weighted	55.9	65.1	54.0	85.9	81.5	93.3

Table 4: Results of the evaluation of the different sentiment analysis approaches. Best results per metric are marked in bold.

NLP approaches

Experiment result	Accuracy in percentage			
	Feature Extraction Techniques			
Algorithms	BOW	TF-IDF	Pre-trained Word2vec	Embedding Layer
SVM	0.78	0.80	0.82	-
NB	0.80	0.80	0.74	-
RF	0.79	0.79	0.81	-
XGBoost	0.80	0.77	0.81	-
CNN	-	-	0.81	0.82
BI-LSTM	-	-	0.84	0.81

Table 5: Eight classes experiment result with classical, ensemble, Deep ML classifier

- **Source:** Ababu, Teshome Mulugeta, and Michael Melese Woldeyohannis. "Afaan Oromo hate speech detection and classification on social media." Proceedings of the thirteenth language resources and evaluation conference. 2022.

Rule-based NLP

- Example one: **Tokenizer**
- **Rule 1:** Replace every punctuation with “white space + punctuation”,
“I like apples.” -> “I like apples .”
- **Rule 2:** Replace every white spaces with newline
- Python implementation “tokenizer.py & tokenizer.ipynb”

Rule-based NLP

- Example two: **Language identifier**
- Step 1: given corpora in different languages, extract most 100 frequent bigrams or trigrams [fleets -> ["fl", "le", "et", "ts"] or ["fle", "lee", "ets"]]
- Step 2: convert the input text into bigrams or trigrams
- Step 3: calculate a score between the bigrams or trigrams of the input text with the most frequent bigrams and trigrams from each language
- Calculate a score of the number of matches and choose the language with the highest score
- Python implementation "lan_identifier.py & lan_identifier.ipynb"

Rule-based NLP

- Example three: [Sentiment analysis](#)
- VADER (Valence Aware Dictionary and sEntiment Reasoner)
- Nothing to do with Star Wars Vader
- Lexicon and rule-based sentiment analysis tool
- Built for sentiment analysis in social media
- Lexicon: (large vocabulary [pos, neg], valence score for each word)
- Rules: (punctuation, capitalization, adverbs usage, conjunction and negation, etc)
- Doesn't generalize well, fail with mixed sentiment
- Example code implementation on [github](#)
- NLTK code implementation on [NLTK](#), NLTK [Tutorial](#) on Sentiment Analysis
- [Code example](#) of using NLTK vader for sentiment analysis

Rule-based NLP

- Discuss in group of 2 - 3 the following (5-10min):

- 1- What are the pros and cons of using such approaches for example 1 & 2?

- 2- Find examples where the tokenizer or the language identifier would fail

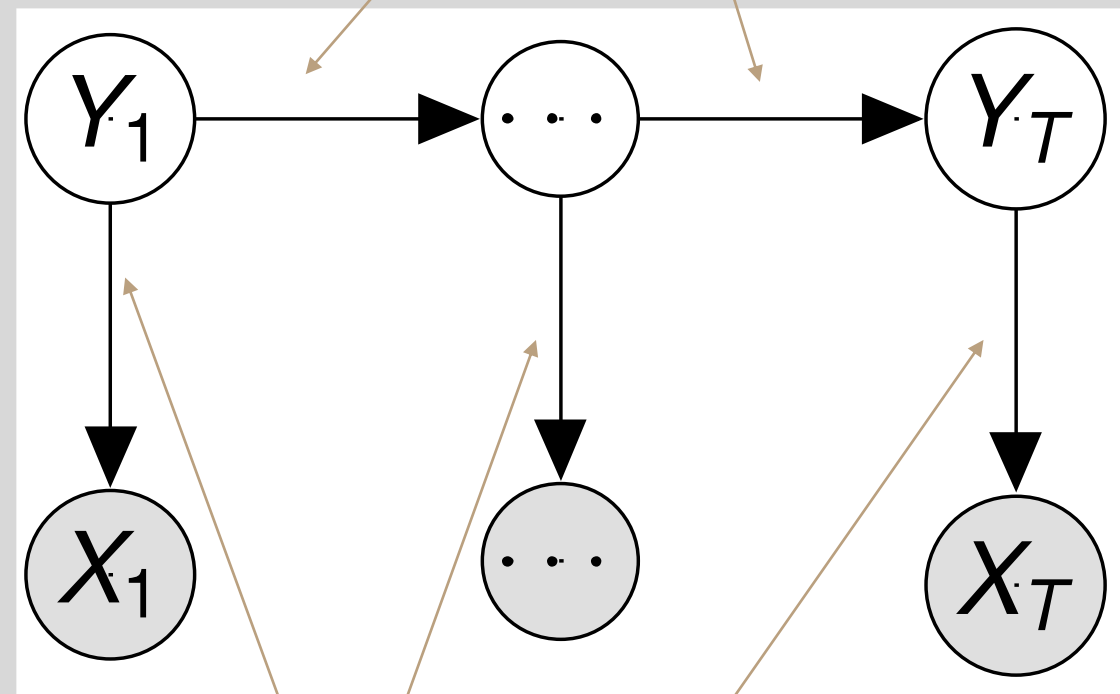
Statistical NLP (HMM)

- Relies heavily on probability theory
- Example: Hidden Markov Model (HMM) for POS tagging
- Given a PoS-tagged training corpus, HMM model calculates the joint probability distribution $P(X, Y)$: What is the probability of observing a sequence of words x with PoS-tag labels y ?
- From this joint probability $P(X, Y)$, we can then infer the conditional probability $P(Y | X)$ that given a certain sequence of words x the correct PoS-tag labels are y by applying Bayes rule.
- After having probability distribution, we chooses the best label sequence with argmax .

Statistical NLP (HMM)

- Observed events x_1, \dots, x_T : words/tokens that we can see in the input
- Hidden events y_1, \dots, y_T : part-of-speech tags that we think of as causal factors for the observed events.
- Assumption 1: Each token only depends on the current part-of-speech
- Assumption 2: Each part-of-speech depends only on the immediately preceding part-of-speech

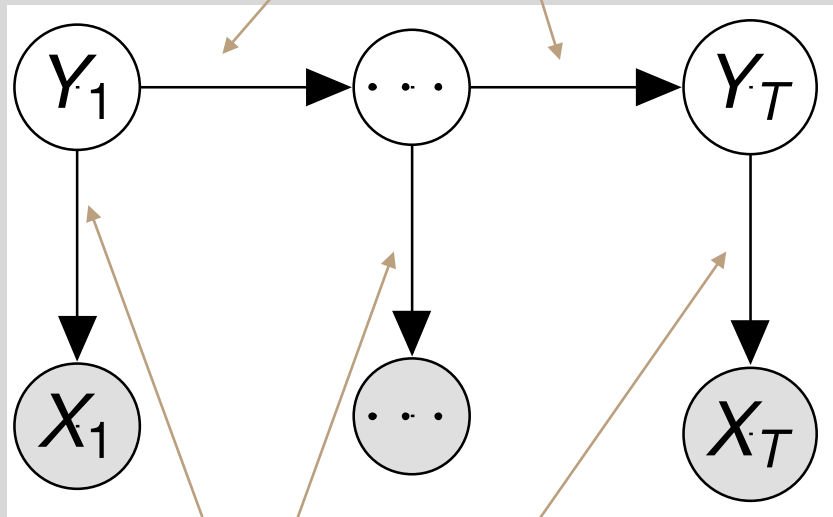
Transition probability $P(Y_i | Y_{i-1})$



Output probability
 $P(X_i | Y_i)$

Statistical NLP (HMM)

Transition probability $P(Y_i | Y_{i-1})$



Output probability
 $P(X_i | Y_i)$

Example: Transition probability

	NNP	MD	VB	JJ	NN	RB	DT
<s>	0.2767	0.0006	0.0031	0.0453	0.0449	0.0510	0.2026
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	0.0090	0.0025
MD	0.0008	0.0002	0.7968	0.0005	0.0008	0.1698	0.0041
VB	0.0322	0.0005	0.0050	0.0837	0.0615	0.0514	0.2231
JJ	0.0366	0.0004	0.0001	0.0733	0.4509	0.0036	0.0036
NN	0.0096	0.0176	0.0014	0.0086	0.1216	0.0177	0.0068
RB	0.0068	0.0102	0.1011	0.1012	0.0120	0.0728	0.0479
DT	0.1147	0.0021	0.0002	0.2157	0.4744	0.0102	0.0017

Figure 8.7 The A transition probabilities $P(t_i | t_{i-1})$ computed from the WSJ corpus without smoothing. Rows are labeled with the conditioning event; thus $P(VB|MD)$ is 0.7968.

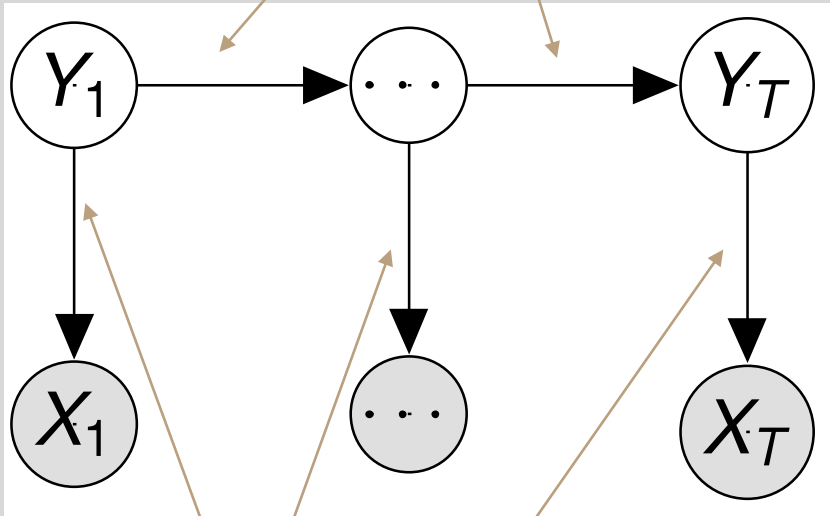
Example: Output probability

	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0	0
NN	0	0.000200	0.000223	0	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0

Figure 8.8 Observation likelihoods B computed from the WSJ corpus without smoothing, simplified slightly.

Statistical NLP (HMM)

Transition probability $P(Y_i | Y_{i-1})$



Output probability
 $P(X_i | Y_i)$

Joint probability that a certain sequence of words x_1, \dots, x_T with PoS-tags y_1, \dots, y_T occurs

$$P(Y_1, \dots, Y_T, X_1, \dots, X_T) = P(Y_1)P(X_1|Y_1) \prod_{t=2}^T P(Y_t|Y_{t-1})P(X_t|Y_t)$$

$P(Y_1)$: Probabilities for the first PoS-tag of a sequence

$P(Y_t|Y_{t-1})$: Transition probabilities: conditional probability of a PoS-tag given the immediately preceding PoS-tag

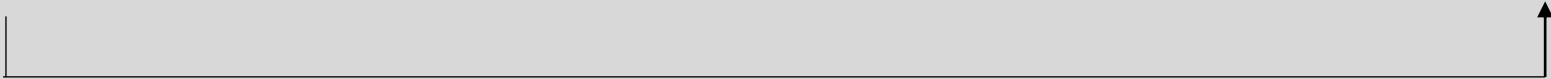
$P(X_t|Y_t)$: Output probabilities conditional on the PoS-tag (including $P(X_1|Y_1)$)

Statistical NLP (HMM)

- After training HMM for POS tagging, we can predict POS tags for a sequence of tokens
- We use argmax function

Example:

Given the token sequence `The man tries` find the most likely sequence of PoS-tags:

$$\underset{y \in \{(DT, NN, VBZ), (NN, VBZ, DT), (DT, NS, NS), \dots\}}{\operatorname{argmax}} P(Y = y \mid X = (\text{The}, \text{man}, \text{tries})) = (DT, NN, VBZ)$$


A diagram consisting of a horizontal line with a vertical line segment at the left end and an upward-pointing arrow at the right end, connecting the set of possible tag sequences to the specific tag sequence (DT, NN, VBZ).

Neural Networks

- On Wednesday with details!