**Slides 2**

**Slides 3**

Ambiguities

Disambiguation deciding whether “make” means “create” or “cook” and can be solved by word sense disambiguation

ASR Automatic speech recognition, problem in phonetics/phonology

Spectogram

WordNet Lemmatizer

Stemming and Lemmatizing

POS

Corpora

ASR how to solve

Spectrogram -> Acoustic Model (convertes to matrics of probabilities) -> Decoder -> punctuation and capitalization model

A group of rectangular boxes with text

AI-generated content may be incorrect.

A diagram of a person's mind

AI-generated content may be incorrect.

Responsible AI rules:

* Diverse and Representative training data
* Bias detection and mitigation
* Privacy by design
* Fact checking and verification
* Explainable AI (XAI)
* User Education

**L2: Basic Text Processing**

**Regular Expressions:**

Closed in slashes **/**…**/**

Whitespace characters **\n** (new line) **\r** (carriage return), **\t** (tab), **\0** (null)

Disjuction (that or that) /**[**Ww**]**…/

Ranges characters /**[0-9][A-Z]**/

Negation [**^**...]

Pipe (or) /…**|**…/

s/(\d+)/\1%/

Optional (previous) /…**s?**.../

One or more /…**s+**…/

Zero or more /…**s\***…/

Ranges repetitions /…h**i{1,3}**/

Any charater /**.**/

Boundries end /**\b**…**\b**/

Replace patterns s/<MATCHING PATTERN>/<REPLACEMENT STRING>/

Capture groups s/**(**\d+**)**.**(**\d+**)**/**\1**,**\2**/ /We **(**.+ed**)** and **\1**/

Lookahead Assertion are patterns that take into account successive

tokens without consuming them.

(?= pattern) is true if the pattern matches

(?! pattern) is true if a pattern does not match.

Corpora (plural of corpus) are large and structured sets of texts, used especially in linguistics and natural language processing (NLP) for analysis, training, or evaluation.

**Tokenization:**

Punctuation serves multiple purposes in a sentence

* Abbreviations: m.p.h., Ph. D.
* Saxon genitive: Joshua’s book
* Numbers: 32.55, $15.23
* Dates: 2025-02-24
* URLs: http://www.google.es/
* Email Addresses: [somebody@somewhere.com](mailto:somebody@somewhere.com)

Clitics words that are grammatically independent but are phonetically tied to other words, like Are in We’re

compound words connecting different words to get a new one with new meaning

**Subword tokenization:**

There are various techniques in vogue:

Byte-Pair Encoding (BPE) (Sennrich et al., 2016)

Unigram Language Modeling Tokenisation (Kudo, 2018)

WordPiece (Schuster and Nakajima, 2012)

They consist of two parts:

A token learner that induces a vocabulary from a corpus

A token segmenter that takes a raw sentence and tokenizes it accordingly

The BPE learner algorithm:

1 Let the initial vocabulary 𝑉 be the set of all individual characters

2 Let 𝐶 be the text corpus

3 for 𝑖 = 1 to 𝑘 do

4 𝑡𝑙, 𝑡𝑟 ← Most Frequent pair of adjacent tokens in 𝐶

5 𝑡new ← 𝑡𝑙 + 𝑡𝑟

6 𝑉 ← 𝑉 + 𝑡new

7 Replace all occurrences of (𝑡𝑙, 𝑡𝑟) in 𝐶 by 𝑡new

8 return 𝑉

**Word Normalization:**

consists on setting words/tokens in a standard format for semantically equivalent terms

with multiple forms.

We could implement asymmetric expansion:

* Enter: window→ window, windows
* Enter: windows → Windows, windows
* Enter: Windows → Windows

… but it is less efficient

**Lemmatisation:**

consists on finding the correct dictionary headword form (lemma, root) for a given word.

* am, are, is → be
* car, cars, car’s, cars’ → car

It is done by Morphological parsing. Morphemes are the small meaningful units that compound into words.

* Stems: The “core” meaning of words
* Affixes: Parts that adhere to stems, usually with grammatical functions

**Stemming**

consists on removing all affixes from words crudely to reduce variability of words. Useful in tasks such as classical machine translation. Less refined and complex than Lemmatisation.

A diagram of a sentence

AI-generated content may be incorrect.

**The Minimum Edit Distance**

between two strings is the minimum number of editing operations required to transform one string into the other.

* Insertions
* Deletions
* Substitutions

A screenshot of a graph

AI-generated content may be incorrect.

**L3: Syntatic Parsing**

**Syntactic Structure**

Two views of linguistic structure:

*Constituency*

phrase structure organizes words into nested constituents that can appear in different places eg. John talked [to the children] [about drugs]

words are given a category (PoS - Part of Speech)the (Det), cat (N), cuddly (Adj), by (P), door (N)

A diagram of a structure

AI-generated content may be incorrect.

*Dependency Structure*

A diagram of a tortoise

AI-generated content may be incorrect.

*Ambiguity -* Two or more possible interpretations: word, phrase, sentence. Can be Lexical (don’t know whether noun or verb), structural, reference or phonological

*Vagueness -* Language contains vague or relative terms

*Imperfection* - Understanding a linguistic context often requires a general background knowledge

*Redundancy -*  communicating the same thing in multiple places like “Yesterday I was in Gothenburg.”

*Code-switching – talking in more than one languages in the same sentence*

**📝 Parsing, Treebanks & Statistical Parsers – Quick Note**

**🔍 Parsing**

* The process of analyzing the **grammatical structure** of a sentence.
* Aims to determine how words group into larger phrases (e.g., noun phrases, verb phrases).
* Output is often a **syntax tree** that shows hierarchical relationships.

**🌳 Treebanks**

* Large annotated datasets where sentences are paired with **syntactic parse trees** (manually or automatically labeled).
* Used for:
  + **Training** and **evaluating** parsers,
  + Conducting linguistic research.
* Examples:
  + **Penn Treebank** (English),
  + **Universal Dependencies (UD)** – multilingual, includes part-of-speech tags and syntactic relations.

**📈 Statistical Parsers**

* Unlike rule-based parsers, these use **probabilities** and **training data** to select the most likely parse.
* Can handle **ambiguous or informal language** by relying on learned patterns.
* Based on models like:
  + **PCFG (Probabilistic Context-Free Grammars)**
  + **Transition-based or graph-based dependency parsers**
* Modern approaches use **deep learning**, e.g., BERT-based parsers with BiLSTMs.

**Grammar G <N, Σ ,P, S>**

A finite set N of non terminal symbols or variables.

A finite set Σ of terminal symbols that are disjoint from N.

A finite set P of production rules of the form (Σ U N)\* N (Σ U N)\* -> (Σ U N)\*

where \* is the Kleene star operator and U denotes the set union. Each production rule

maps from one string of symbols to another where the left hand side contains at least

one non terminal symbol.

A distinguished start symbol S ∈ N.

*Regular Grammar*

* A regular language is generated by a regular grammar.
* A grammar is regular if productions are right-linear or left-linear
* V → V T | T (left-linear grammar, T = terminal)
* V → T V | T (right-linear grammar , T = terminal)
* VERY restricted grammar, Unambiguous
* Regular grammar is a subset of context-free grammar

*Context-free Grammar G = (T, N, S, R)*

* A context-free language is generated by a context-free grammar (any combination of terminals and non-terminals)
* A grammar G <N, Σ, P, S> is context-free if the production rules are of the form N → (N U Σ)\*
* Less restricted than Regular Grammar
* Different context-free grammars can generate the same context-free language
* They can be Unambiguous (only one parse tree) or Ambiguous (more than one parse tree)

**Phrase structure grammars in NLP**

G = (T, C, N, S, L, R)

* T is a set of terminal symbols
* C is a set of preterminal symbols
* N is a set of nonterminal symbols
* S is the start symbol (S ∈ N)
* L is the lexicon, a set of items of the form X → x
* X ∈ P and x ∈ T
* R is the grammar, a set of items of the form X → 
* X ∈ N and  ∈ (N ∪ C)\*

**Probabilistic (or stochastic) context-free grammars (PCFGs)**

T is a set of terminal symbols

N is a set of nonterminal symbols

S is the start symbol (S ∈ N)

R is a set of rules/productions of the form X → 

P is a probability function

**Restricting the grammar form for efficient parsing Chomsky Normal Form**

* We should think of this as a transformation for efficient parsing
* In practice full Chomsky Normal Form is a pain
  + Reconstructing n-aries is easy
  + Reconstructing unaries/empties is trickier

**BinarizationA diagram of a tree

AI-generated content may be incorrect.**

**CKY ParsingA screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Other parsings:**

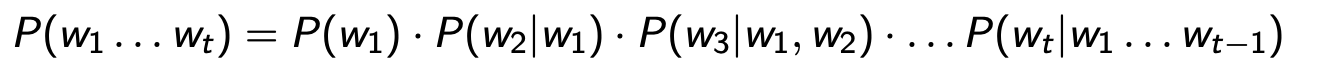
* MaltParser
* Graph-based dependency parser

**L4: Language models**

**Text generating**

Linters - to narzędzia, które analizują kod źródłowy i znajdują potencjalne błędy, problemy ze stylem

A Language Model is function that assigns a probability to a sequence of words. Some applications also use the concept of conditional probabilities to model the probability of the next word given a sequence.

Markov Assumption

n-gram is a sequence of n words.

n-gram model computes the probability of the n + 1th word given the previous n.

Laplace Smoothing – when words that are present in the vocabulary but appear in the test set in an unseen context, “Remove” some probability from the most likely n-grams and give it to them

Backoff - if we want to look for a trigram P (wk |wk−2,wk−1) which is not available, resort to using the bigram P (wk |wk−1). Keep reducing the n until a viable n-gramis found or until n = 1

Stupid backoff – when backing off multiply by something to decrease the probability

**Feed-Forward Neural LM**

Advantages

* Can handle much longer histories (defined by length of network input)
* Can generalise better over contexts of similar words
* More accurate at word prediction.

Disadvantages

* More complex.
* Need to be trained.
* Lack of explainability.
* No arbitrary length input.
* The larger the window, the larger vocabulary can become the problem
* Vocabulary size can become a problem
* Each wor
* d is multiplied by a different set of weights in W, meaning their position matters

Perplexity (PP, PPL) is a measure of how well a LM fits a set of unseen data. The best LM is the one which gives the most probability to the samples in the test set. P(W ) →1 ⇒PPL(W ) →1 and for a P(W ) →0 ⇒PPL(W ) →∞

**L5: Sequence Labelling**

**Sequence labelling**

Examples of sequence labelling tasks:

• Part-of-Speech (POS)

• Named Entity Recognition (NER)

Examples of non-sequence labelling tasks (size input ≠ size output):

• Translation

• Summarization

**Parts of Speech (POS)**

noun, verb, pronoun, preposition, adverb, conjunction, participle, article

A screenshot of a diagram

AI-generated content may be incorrect.

A blue table with black text

AI-generated content may be incorrect.A blue and black text with black text

AI-generated content may be incorrect.

Supervised Machine Learning Algorithms:

• Hidden Markov Models (HMMs)

• Conditional Random Fields (CRF)/ Maximum Entropy Markov Models (MEMM)

• Neural sequence models (RNNs or Transformers)

• Large Language Models (like BERT), finetuned

**Named Entity Recognition (NER):**

Named entity, in its core usage, means anything that can be referred to with a proper name.

Most common 4 tags:

* PER (Person): “Marie Curie”
* LOC (Location): “Palma de Mallorca”
* ORG (Organization): “Universitat Autònoma de Barcelona”
* GPE (Geo-Political Entity): “Castilla la Mancha"

Often multi-word phrases

But the term is also extended to things that aren't entities: Dates, times, prices

When to use?

* Sentiment analysis: consumer’s sentiment toward a particular company or person?
* Question Answering: answer questions about an entity?
* Information Extraction: Extracting facts about entities from text.

BIO Tagging:

* B: token that begins a span
* I: tokens inside a span
* O: tokens outside of any span

Standard algorithms for NER:

• Hidden Markov Models (HMM)

• Conditional Random Fields (CRF)/ Maximum Entropy Markov Models (MEMM)

• Neural sequence models (RNNs or Transformers)

• Large Language Models (like BERT), finetuned

POS taggers are evaluated by the standard metric of Accuracy.

NER taggers are evaluated by Recall, Precision, and F1 measure.

**HMMs:**

A math equation with black text

AI-generated content may be incorrect.

A Markov chain makes a very strong assumption that if we want to predict the future in the sequence, all that matters is the current state. All the states before the current state have no impact on the future except via the current state

HMM has two componentsA math equations and formulas

AI-generated content may be incorrect.A math equations with numbers and a red circle

AI-generated content may be incorrect.A diagram of a graph

AI-generated content may be incorrect.