Computational Argumentation – Part II

Basics of Natural Language Processing

Henning Wachsmuth

https://ai.uni-hannover.de

Leibniz Universität Hannover

Learning goals

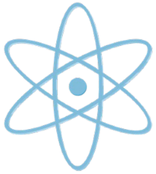
- **Concepts**
  - Basics from linguistics, statistics, and machine learning

https://commons.wikimedia.org

- **Methods**
  - How to develop and evaluate data-driven algorithms
  - Standard techniques used in machine learning
  - Types of analyses used in natural language processing (NLP)

https://pixabay.com

- **Associated research fields**
  - Natural language processing



https://pixabay.com

- **Within this course**
  - Concepts and methods this course builds upon

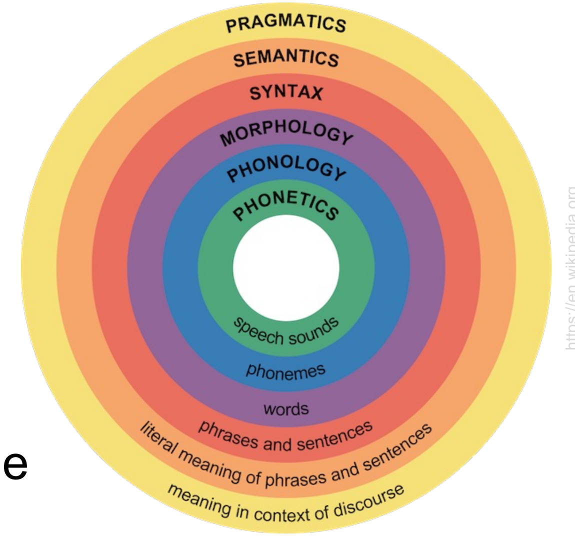
- **Disclaimer**
  - The basics selected here are all but complete and only revisited high-level

    For more details, see e.g. my old bachelor‘s course: https://cs.upb.de/css/teaching/courses/text-mining-w20

Outline

What is linguistics?

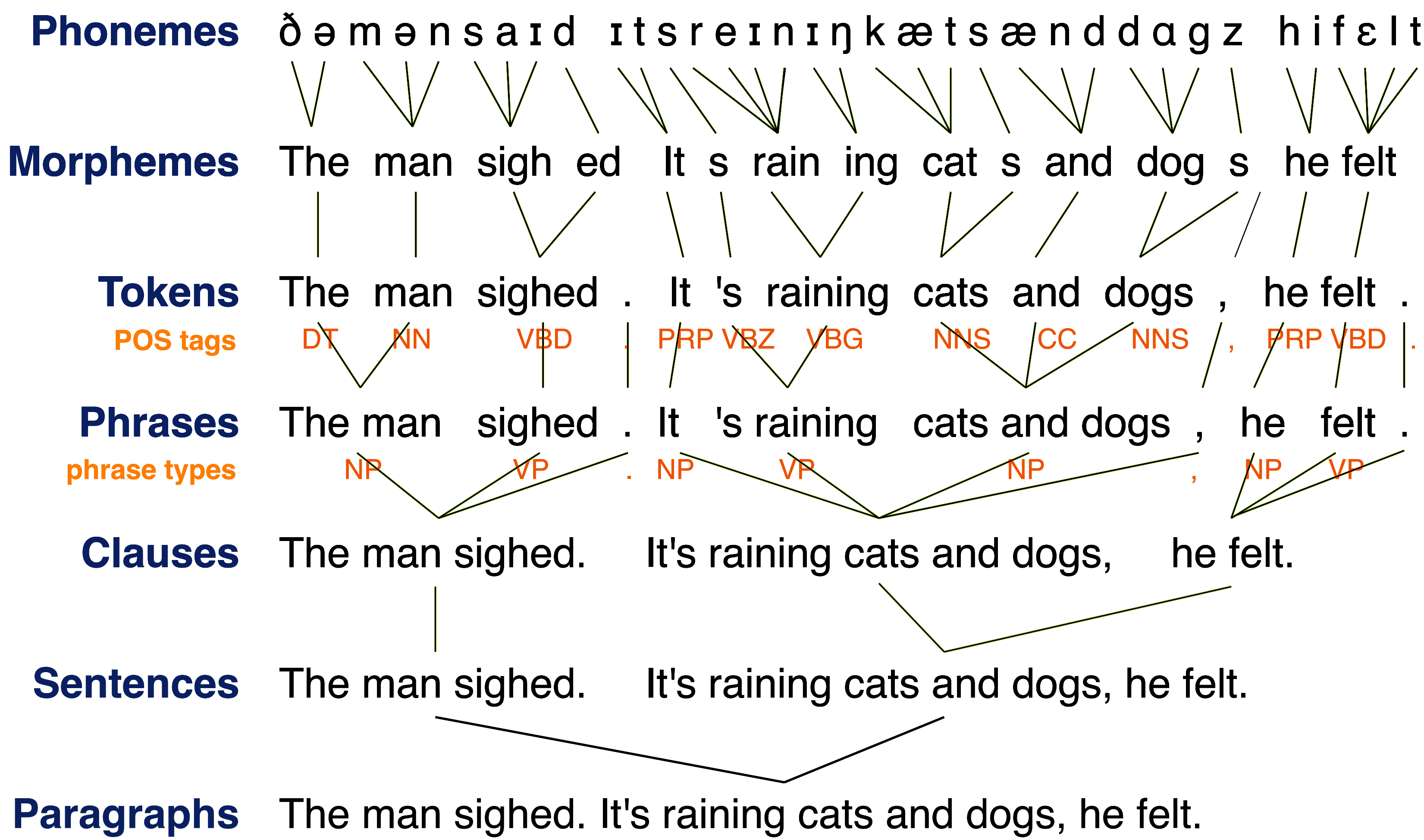
- **Linguistics**
  - The study of spoken and written natural language in terms of the analysis of form, meaning, and context



- **Levels of spoken language only**
  - Phonetics. Physical aspects of speech sounds
  - Phonology. Linguistic sounds of a particular language

- **Levels of spoken and written language**
  - Morphology. Senseful components of words and wordforms
  - Syntax. Structural relationships between words, usually in a sentence
  - Semantics. Meaning of single words and compositions of words
  - Discourse. Linguistic units larger than a single sentence
  - Pragmatics. How language is used to accomplish goals

Linguistic text units



Main morphological concepts

- **Word**
  - The smallest unit of language that is to be uttered in isolation
    Example: ”cats“ and ”ran“ in ”cats ran.“
- **Lemma**
  - The dictionary form of a word
    Example: ”cat“ for ”cats“, ”run“ for ”ran“
- **Stem**
  - The part of a word(form) that never changes
    Example: ”cat“ for ”cats“, ”ran“ for ”ran“
- **Token**
  - The smallest text unit in NLP: A wordform, number, symbol, or similar
    Example: ”cats“, ”ran“, and ”.“ in ”cats ran.“ (whitespaces are not considered as tokens)

Main syntactic concepts

- **Part-of-speech (POS)**
  - The lexical category (or word class) of a word
  - Abstract classes. Nouns, verbs, adjectives, adverbs, prepositions, ...
  - POS tags. NN (single nouns), NNS (plural nouns), NNP (proper nouns), ...
- **Phrases**
  - A contiguous sequence of related words, functioning as a single meaning unit
  - Phrases often contain nested phrases.
  - Types. Noun phrase (NP), verb phrase (VP), prepositional phrase (PP)
    Sometimes also adjectival phrase (AP) and adverbial phrase (AdvP)
- **Clause**
  - The smallest grammatical unit that can express a complete proposition
  - Types. Main clause and subordinate clause
- **Sentence**
  - A grammatically independent linguistic unit consisting of one or more words

Main semantic concepts

- **Lexical semantics**
  - The meaning of words and multi-word expressions
    Different senses of a word, the roles of predicate arguments, ...
- **Compositional semantics**
  - The meaning of the composition of words in phrases, sentences, and similar
    Relations, scopes of operators, and much more
- **Entities**
  - An object from the real world
  - Named entities. Persons, locations, organizations, products, ...
    For example, ”Prof. Dr. Henning Wachsmuth”, “Hannover”, “Leibniz University Hannover”
  - Numeric entities. Values, quantities, ranges, periods, dates, ...
    For example, “in this year”, “2023-04-25”, “$ 100 000”, “762-123 77”
- **Relations**
  - Semantic. Relations between entities, e.g., organization *founded in* period
  - Temporal. Relations describing courses of events, e.g., as in news reports

Main discourse and pragmatics concepts

- **Discourse**
  - Linguistic utterances larger than a sentence, e.g., paragraphs or entire texts
    Usually monological; dialogical discourse is rather referred to as *dialogue*.

- **Discourse structure**
  - Discourse segments. Building block of a discourse in terms of linguistic units
  - Coherence relations. Semantic or pragmatic relations between segments
    Examples: A causes B, B elaborates A, A contrasts to B, …

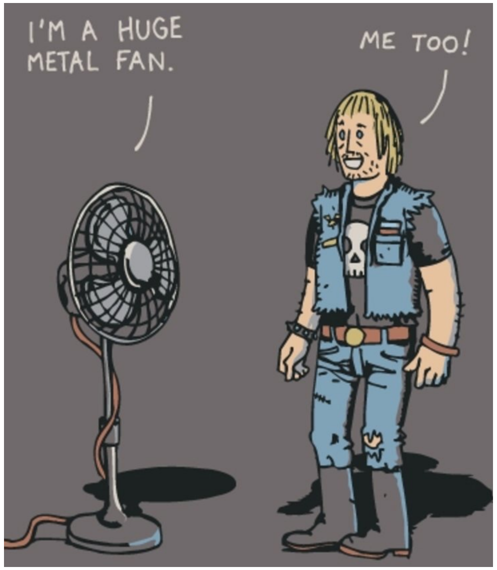
- **Coreference**
  - Two or more expressions in a text that refer to the same thing
  - Types. Pronouns in anaphora and cataphora, coreferring noun phrases, ...
    Examples: ”**Apple** is based in Cupertino. **The company** is actually called **Apple Inc.**, and **they** make hardware.“

- **Speech acts**
  - Linguistic utterances with a performative function
  - Speakers may commit to something, make the listener do something, …
    Example: ”Smoking is bad for your health“ aims to make you not smoke.

What makes language understanding hard?

- **Ambiguity**
  - The fundamental challenge of NLP is that language is ambiguous

- **Ambiguity is pervasive**
  - Phonetic. ”wreck a nice beach”
  - Word sense. ”I went to the bank.”
  - Part of speech. ”I made her duck.”
  - Attachment. ”I saw a kid with a telescope.”
  - Coordination. ”If you love money problems show up.“
  - Scope of quantifiers. ”I didn’t buy a car.”
  - Speech act. “Have you emptied the dishwasher?”



- **Other challenges**
  - World knowledge. ”Putin must rethink his view of Ukraine”
  - Domain dependency. ”Read the book!”

  ... and many more

Next section: Empirical methods

I. Introduction to computational argumentation

II. Basics of natural language processing

a) Linguistics

b) Empirical methods

c) Tasks and techniques

d) Rule-based NLP

e) Statistical NLP

f) Conclusion

III. Basics of argumentation

IV. Argument mining

V. Argument assessment

VI. Argument generation

VII. Applications of computational argumentation

VIII. Conclusion

Development and evaluation in NLP

- **Development and evaluation**
  - NLP algorithms are developed based on *text corpora*.
  - Their output is rarely free of errors, which is why it is usually evaluated empirically in comparison to ground-truth annotations.

- **Evaluation criteria**
  - Effectiveness. The extent to which the output of an algorithm is correct
  - Efficiency. The consumption of time (or space) of an algorithm on an input
  - Robustness. The extent to which an algorithm remains effective across different inputs, often in terms of textual domains

- **Evaluation measures**
  - Quantify the quality of an algorithm on a specific task and text corpus
  - Algorithms can be ranked with respect to an evaluation measure.
  - Different measures are useful depending on the task.

Annotated text corpora

- **Text corpus**
  - A collection of real-world texts with known properties, compiled to study a language problem
  - The texts are often *annotated* with meta-information.
  - Corpora are split into datasets for developing and/or evaluating (testing) an algorithm.

- **Annotations**
  - Marks a text or span of text as representing meta-information of a specific type
  - Labels or numeric values
  - Also used to specify relations

Time entity **Organization entity**
" 2014 ad revenues of Google are going to reach

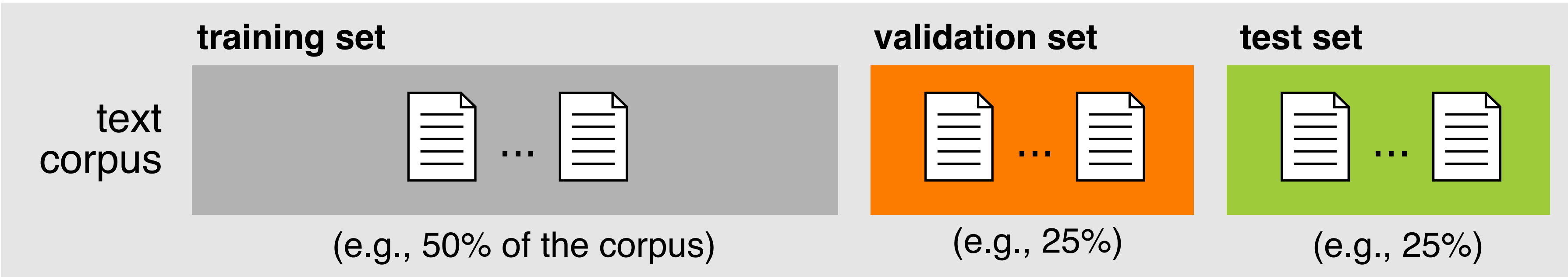
Reference **Time entity**
$20B. The search company was founded in '98.

Reference **Time entity** **Founded relation**
Its IPO followed in 2004. [...] "

Topic: "Google revenues" **Genre:** "News article"

- **Types of annotations**
  - Ground-truth. Manual annotations, often created by experts
  - Automatic. NLP algorithms add annotations to texts.

Training, validation, and test set



- **Training set**
  - Known instances used to develop or statistically learn an algorithm
  - The training set may be analyzed manually and automatically.

- **Validation set** (aka development set)
  - Unknown test instances used to iteratively evaluate an algorithm
  - The algorithm is optimized towards and adapts to the validation set.

- **Test set** (aka held-out set)
  - Unknown test instances used for the final evaluation of an algorithm
  - The test set represents unseen data.

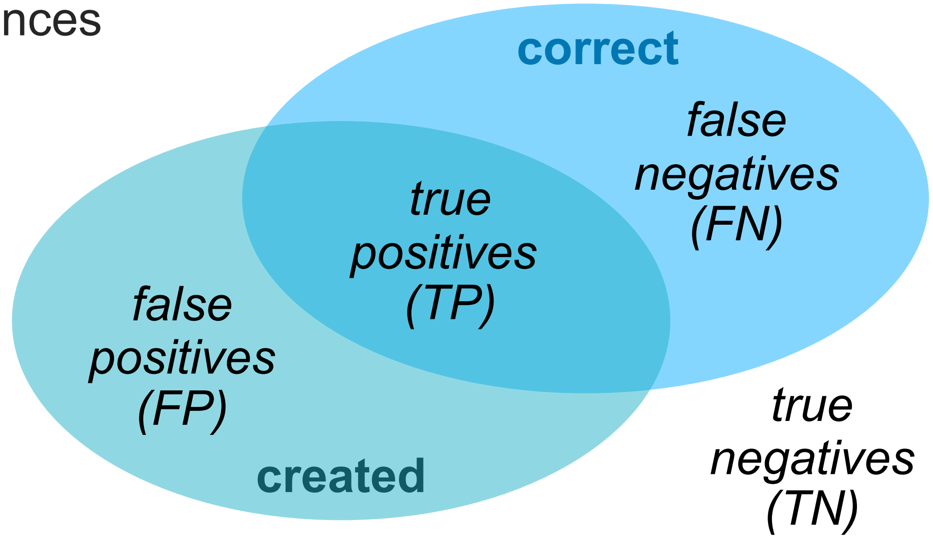
Cross-validation

- **(Stratified) *n*-fold cross-validation**
  - Randomly split a corpus into *n* datasets of equal size, usually *n = 10*
  - The development and evaluation consist of *n* runs. The evaluation results are averaged over all *n* runs.
  - In the *i*-th run, the *i*-th fold is used for evaluation (testing). All other folds are used for development (training).

- **Pros and cons of cross-validation**
  - Often preferred when data is small, as more data is given for training
  - Cross-validation avoids potential bias in a corpus split.
  - Random splitting often makes the task easier, due to corpus bias.

Evaluation of effectiveness in predicting nominal labels

- **Nominal instances (labels)**
  - Positives. The output instances (annotations) an algorithm has created
  - Negatives. All other possible instances



- **Accuracy**
  - Used if positives and negatives are similarly important

$$\textit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision, recall, and F_1-score**
  - Used if positives are in the focus

$$\textit{Precision}\ (P) = \frac{TP}{TP + FP} \qquad \textit{Recall}\ (R) = \frac{TP}{TP + FN} \qquad F_1\textit{-score} = \frac{2 \cdot P \cdot R}{P + R}$$

  - In multi-class tasks, *micro-* and *macro-averaged* values can be computed.

Evaluation of effectiveness in predicting numeric values

- **Numeric instances**
  - NLP algorithms may predict values y_i from a real-valued scale.
  - The numeric difference to the ground-truth values y_i^* is usually in the focus.

- **Mean absolute error (MAE)**
  - Used if outliers require no special treatment

$$MAE = \frac{1}{n} \cdot \sum_{i=1}^{n} |y_i - y_i^*|$$

- **Mean squared error (MSE)**
  - Used if outliers are considered particularly problematic

$$MSE = \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - y_i^*)^2$$

- **Root mean squared error (RMSE)**
  - Just a different way of quantifying the squared error, $RMSE = \sqrt{MSE}$

Comparison

- **Need for comparison**
  - It is unclear how good a measured effectiveness result in a given task is.
  - Comparison against baselines is needed

- **Baseline** (aka lower bound)
  - An alternative approach proposed before or can be developed easily.
  - A new algorithm aims to be better than all relevant baselines.

- **Types of baselines**
  - Trivial. An approach that can easily be derived from a given task or dataset
  - Standard. An approach that is often used for related tasks
  - Sub-approach. A sub-part of a new approach
  - State of the art. The best published approach for the addressed task

Empirical research and variables

- **Empirical methods**
  - Quantitative methods based on numbers and statistics
  - Study questions on behaviors and phenomena by analyzing data
  - Asks about the relationships between *variables*

- **Variable**
  - An entity that can take on different nominal or numeric values
  - Independent. A variable *X* that is expected to affect another variable
  - Dependent. A variable *Y* that is expected to be effected by others
  - Other. Confounders, mediators, moderators, ...

- **Scales of variables**
  - Nominal. Values that represent discrete, separate categories (labels)
  - Ordinal. Values that can be ordered/ranked by what is better
  - Interval. Values whose difference can be measured
  - Ratio. Interval values that have an absolute zero

Descriptive statistics

- **Descriptive statistics**
  - Measures for summarizing and comprehending distributions of values
  - Used to describe phenomena

- **Measures of central tendency**
  - Mean. The arithmetic average of a sample from a distribution of values
  - Median. The middle value of the ordered values in a sample
  - Mode. The value with the greatest frequency in a sample

- **Measures of dispersion**
  - Range. The distance between minimum and maximum in a sample
  - Variance. The mean squared difference between each value and the mean
  - Standard deviation. The square root of the variance

Inferential statistics

- **Inferential statistics**
  - Procedures that study hypotheses based on values
  - Used to make inferences about a distribution beyond a given sample

- **Two competing hypothesis**
  - Research hypothesis (*H*). Prediction about how some inpedendent variables will affect a dependent variable
  - Null hypothesis (H_0). Antithesis to *H*

"The accuracy of our approach is not higher with POS tags than without."

- **Hypothesis test** (aka statistical significance test)
  - A statistical procedure which determines the probability (*p*-value) that results supporting *H* are due to chance (or sampling error)
  - Significance given, if p is ≤ some significance level α (usually 0.05 or 0.01)

- **Steps in a hypothesis test**
  - State *H* and H_0, choose α.
  - Compute *p*-value with an adequate test. Decide whether H_0 can be rejected.

Next section: Task and techniques

Common text analyses

- **Lexical and syntactic**
  - Tokenization
  - Sentence splitting
  - Paragraph detection

  - Stemming
  - Lemmatization
  - Part-of-speech tagging

  - Phrase chunking
  - Dependency parsing
  - Constituency parsing

  ... and some more

- **Semantic and pragmatic**
  - Attribute extraction
  - Numeric entity recognition
  - Named entity recognition

  - Reference resolution
  - Entity relation extraction
  - Discourse parsing

  - Topic detection
  - Sentiment analysis
  - Spam detection

  ... and many many more

Example task: Information extraction

- **Information extraction**
  - Mining of entities, their attributes and relations, and events they participate in from natural language text
  - The output is structured information that can, e.g., be stored in databases.

- **Example task**
  - Extraction of the founding dates of companies

Output: Founded("Google", 1998)

- **Typical text analysis steps**
  1. Lexical and syntactic preprocessing
  2. Named and numeric entity recognition
  3. Reference resolution
  4. Entity relation extraction

Text analysis pipelines and alternatives

- **Text analysis pipeline**
  - The traditional way to tackle an NLP task is with a pipeline that sequentially applies a set of algorithms to the input texts.
  - The output of one algorithm is the input to the next.

- **Example pipeline**
  - Extraction of the founding dates of companies

input text → Sentence splitting → Tokenization → Part-of-spech tagging → Phrase chunking →

Time entity recognition → Named entity recognition → Reference resolution → Founded rela-tion detection → **founded relations**

- **Alternatives**
  - Joint model. Realizes multiple analysis steps at the same time
  - Neural network. Often works on the raw input text (details in this course)

Dimensions of NLP tasks

- **Types of tasks**
  - Classification. Each input instance is assigned a predefined nominal label.
  - Regression. Each input instance is assigned a numeric value.
  - Clustering. A set of input instances is grouped into not-predefined classes.

    ... and some others

- **Types of approaches**
  - Supervised. Training instances with known output used in development
  - Unsupervised. No output labels/values used in development

    ... and some others

- **Types of techniques**
  - Rule-based. Analysis based on manually encoded expert knowledge

    Knowledge includes rules, lexicons, grammars, ...
  - Feature-based. Analysis based on statistical patterns in text features

    Text features are encoded manually or semi-automatically
  - Neural. Analysis based on statistical patterns in self-learned functions

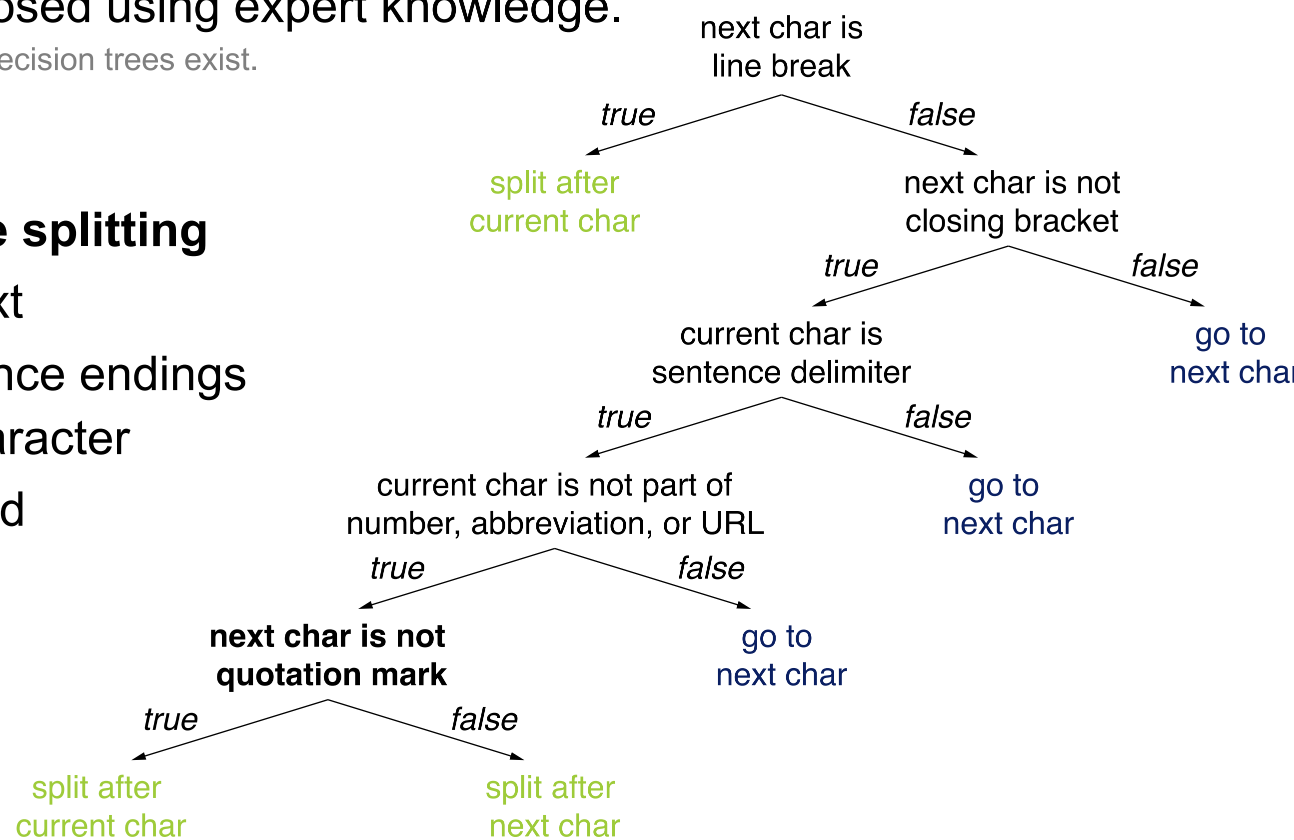
    Complex functions are learned automatically.

Next section: Rule-based NLP

NLP using decision trees

- **(Hand-crafted) Decision trees**
  - The representation of a series of if-then-else decision rules
  - Inner nodes are decision criteria, leafs the final outcomes in a task.
  - Rules are composed using expert knowledge.
    Also, machine-learned decision trees exist.

- **Example: Sentence splitting**
  - Given a plain text
  - Check for sentence endings character by character
  - Split and proceed



NLP using lexicons

- **Several types of lexicons**
  - Terms. Term lists, language lexicons, vocabularies
  - + Definitions. Dictionaries, glossaries, thesauri
  - + Structured information. Gazetters, frequency lists, confidence lexicons
- **Use cases of lexicons**
  - A given lexicon can be used to find all term occurrences in a text.
  - The existence of a given term in a lexicon can be checked.
  - The density or distribution of a vocabulary in a text can be measured.
- **Example: Attribute extraction**
  - Given a training set where attributes are annotated
  - Compute confidence of each term, i.e., how often it is annotated as attribute
  - Consider terms with confidence above a certain threshold as attributes

Attribute	Confidence
minibar	1.00
towels	0.97
wi-fi	0.83
front desk	0.74
alcohol	0.5
waiter	0.4
buffet	0.21
people	0.01

NLP using regular expressions

- **Regular expression** (aka regex)
  - A representation of a regular grammar
  - Combines characters and meta-characters to generalize over language structures
  - Used in NLP mainly to match text spans that follow clear sequential patterns

- **Types of patterns in regexes**
  - Disjunctions. Alternative options, such as `([Ww]oodchuck|[Gg]roundhog)`
  - Negation+choice. Restrictions and arbitrary parts, such as `[^A-Z]` or `19..`
  - Repetitions. Parts that are optional and/or may appear multiple times, such as `woo(oo)?dchuck`, `woo(oo)*dchuck`, or `woo(oo)+dchuck`

- **Example**
  - `(0?[1-9]|[10-31])\.(0?[1-9]|[10-12])\.(19|20)[0-9][0-9]`
    matches German dates, such as 8.5.1945 or 25.04.2023

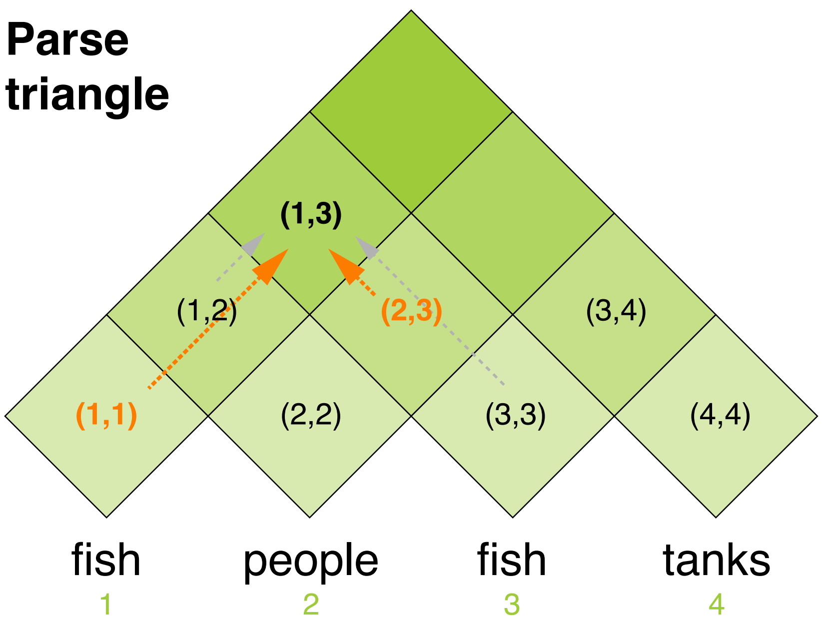
NLP using probabilistic context-free grammars

- **Probabilistic context-free grammar (PCFG)**
  - A CFG where each rule is assigned a probability
  - Used in NLP mainly to parse sentence structure
  - The goal is to find the most likely parse tree

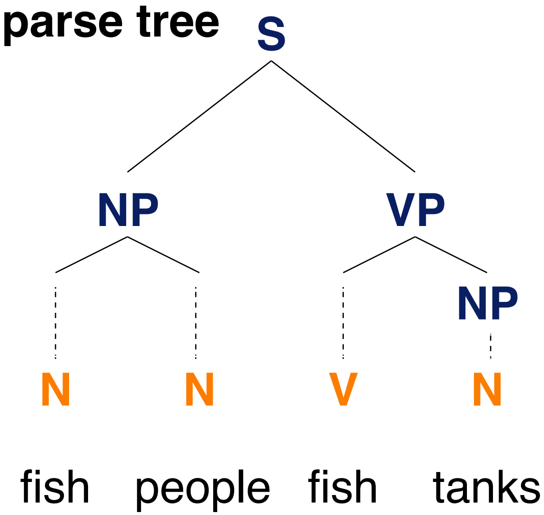
Rule	Probability
S → NP VP	1.0
VP → V NP	0.6
VP → V NP PP	0.4
...	...
V → fish	0.6
V → tanks	0.3

- **Example: Constituency parsing**
  - Dynamic programming to iteratively compute the most likely parse tree

Parse triangle



Most likely parse tree



NLP using similarity measures

- **Similarity measure**
  - A real-valued function that quantifies how similar two instances of the same concept are (between 0 and 1)
  - Distance measures can be used as (inverse) similarity measures.

- **Selected use cases in NLP**
  - Clustering
  - Spelling correction
  - Retrieval of relevant web pages
  - Paraphrase or plagiarism detection

- **Text similarity measures**
  - Vector-based measures. Mainly for similarities between feature vectors
  - Edit distance. For spelling similarities
  - Distributional similarity. For similarities in the contextual usage

Next section: Statistical NLP

Machine learning in NLP

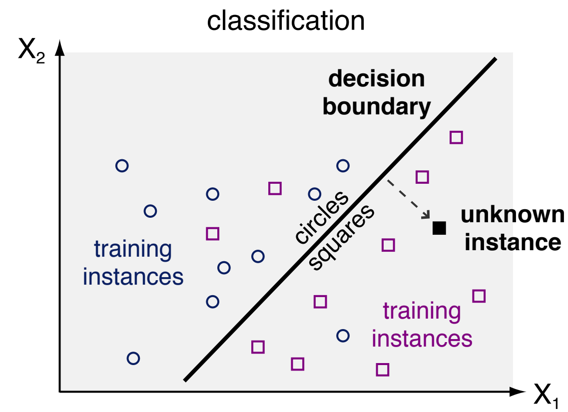
- **Machine learning**
  - An algorithm learns from experience wrt. a task and a performance measure, if its performance on the task increases with the experience.
  - Aims at tasks where a target function γ that maps input to output is unknown
  - A model y is learned that approximates γ

- **Output in NLP**
  - Text labels, such as topic, genre, and sentiment
  - Span annotations, such as tokens and entities
  - Span classifications, such as part-of-speech tags and entity types
  - Relations between annotations, such as entity relations
  - Text generated, such as the answers of ChatGPT

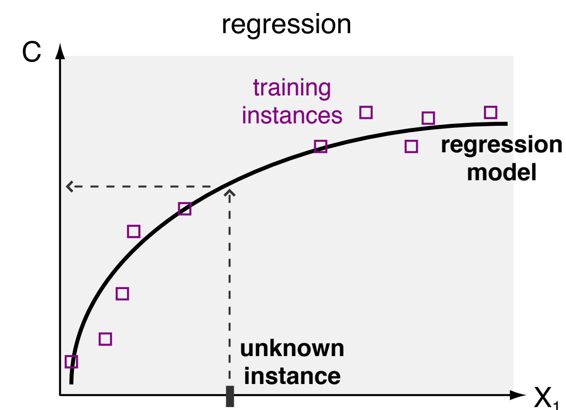
- **Two-way relationship**
  - The output information of NLP serves as input to machine learning
  - Many NLP algorithms rely on machine learning to produce output information

Supervised learning

- **Supervised (machine) learning**
  - A learning algorithm derives a model *y* from known training data, i.e., pairs of instances $x^{(i)}$ and the associated output information $y^{(i)}$.
  - *y* can then predict output information for unknown data.



- **Classification**
  - Assign an instance to the most likely class of a set of predefined classes
  - A decision boundary *y* is learned that decides the class of unknown instances.

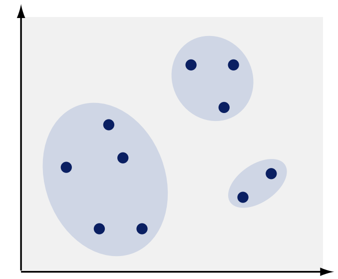


- **Regression**
  - Assign an instance to the most likely value of a continuous target variable
  - A regression function *y* is learned that decides the value of unknown instances.

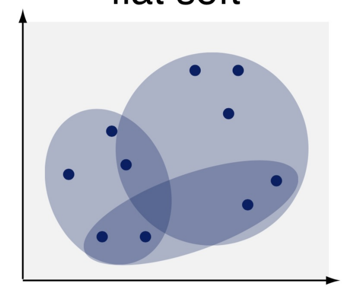
Unsupervised learning

- **Unsupervised (machine) learning**
  - A model *y* is derived from instances without output information.
  - The model reveals the organization and association of data.
- **Clustering**
  - The grouping of a set of instances based on similarity into a possibly but not necessarily predefined number of classes
  - The meaning of a class is usually unknown in advance.
- **Hard vs. soft clusters**
  - Hard. Each instance belongs to a single cluster.
  - Soft. Instances belong to each cluster with a certain weight.
- **Flat vs. hierarchical clustering**
  - Flat. Group instances into a set of independent clusters.
  - Hierarchical. Create a binary clustering tree over all instances.

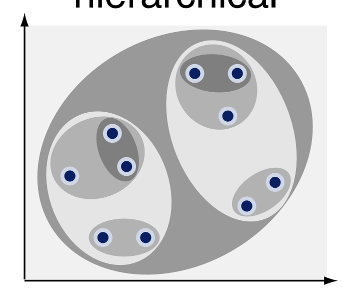
flat hard



flat soft

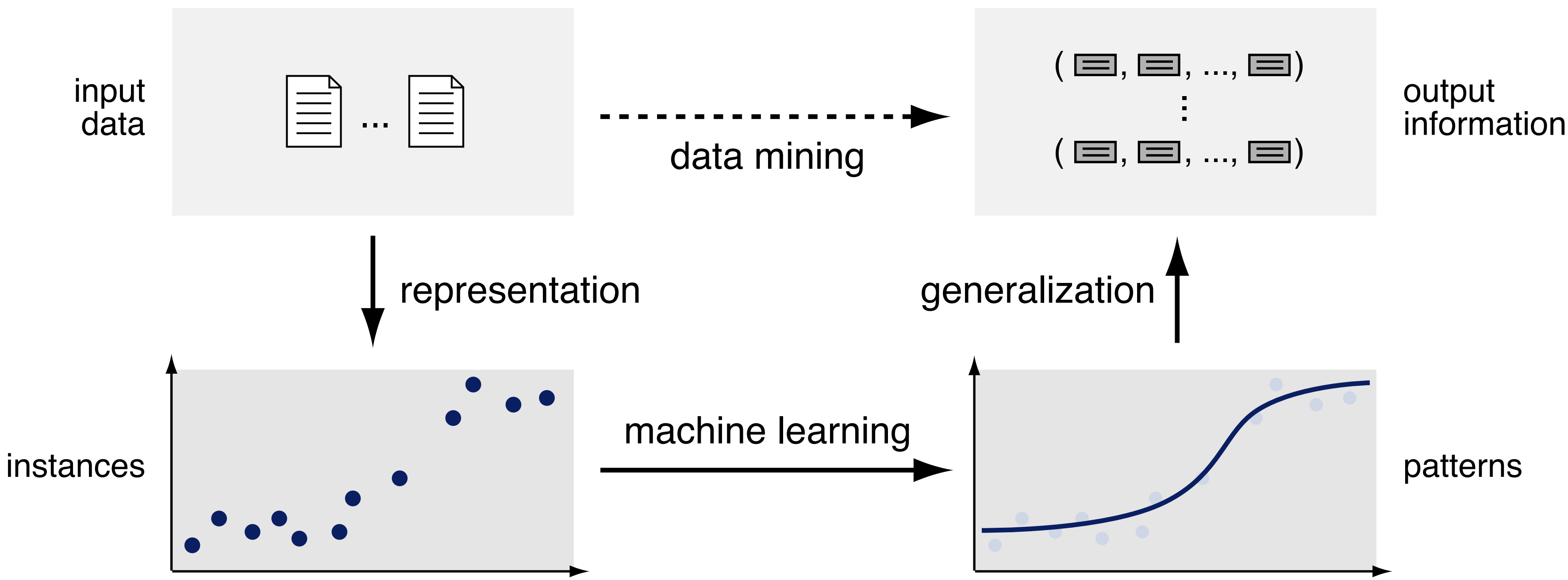


hierarchical



Data mining

- **Data mining vs. machine learning**
  - Data mining puts the output into the view, machine learning the method



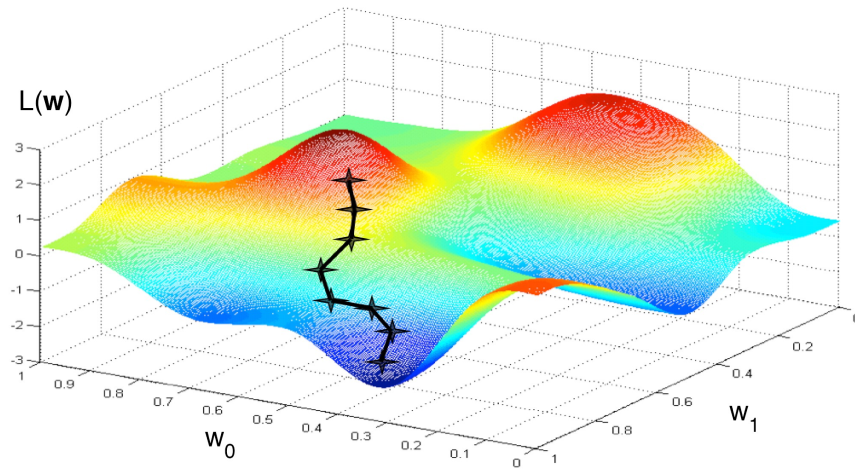
- **NLP as data mining**
  - Input data. A text corpus, i.e., a collection of texts to be processed
  - Output information. Annotations of the texts, or new texts

Representation

- **Feature**
  - A feature *x* denotes any measurable property of an input.
    Example: The relative frequency of a particular word in a text
- **Feature value**
  - The value of a feature of a given input, usually real-valued and normalized
    Example: The feature representing ”is“ would have the value 0.5 for the sentence ”is is a word“.
- **Feature type**
  - A set of features that conceptually belong together
    Example: The relative frequency of each known word in a text (this is often called ”bag-of-words“)
- **Feature vector**
  - A vector $\mathbf{x}^{(i)} = (x_1^{(i)}, ..., x_m^{(i)})$ where each $x_j^{(i)}$ is the value of one feature x_j
    Example: For two feature types with k and l features respectively, $\mathbf{x}^{(i)}$ would contain m = k+l values.
- **Feature-based vs. neural representations**
  - In feature-based learning, each instance is represented as a feature vector.
  - In neural learning, features are not represented explicitly. (more in this course)

Machine learning

- **Machine learning process**
  - A learning algorithm explores several candidate models *y*.
  - Each *y* assigns one weight w_j to each feature x_j.
  - *y* is evaluated on training data against a cost function *L*.
  - Based on the result, the weights are adapted to obtain the next model.
  - The adaptation relies on an optimization procedure.

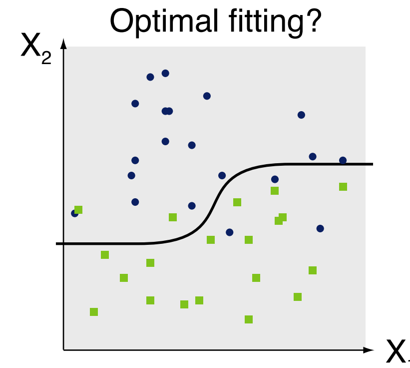


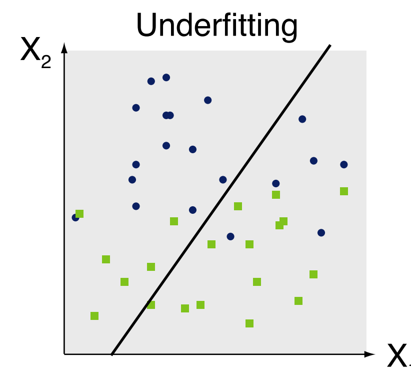
- **Common optimization procedures**
  - Batch gradient descent. In each step, *y* is adapted to all training instances
  - Stochastic gradient descent. Adapts *y* iteratively to each single instance

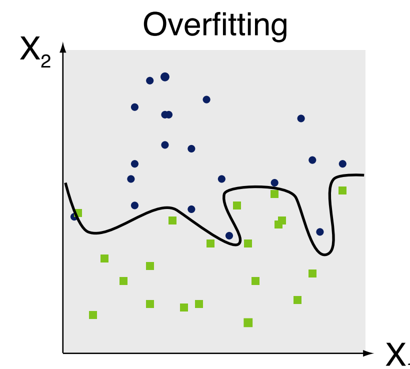
- **Hyperparameters**
  - Many learning algorithms have parameters that are not optimized in training.
  - These hyperparameters need to be optimized against a validation set.

Generalization

- **Fitting**
  - To generalize well, y should approximate the complexity of the unknown function γ based on the training data.
- **Underfitting (too high bias)**
  - The model generalizes too much, not capturing certain relevant properties.
- **Overfitting (too high variance)**
  - The model captures too many irrelevant properties of the input data.
- **Regularization**
  - To avoid overfitting, the use of complex functions can be penalized.
  - A term is added to the cost function that forces feature weights to be small.



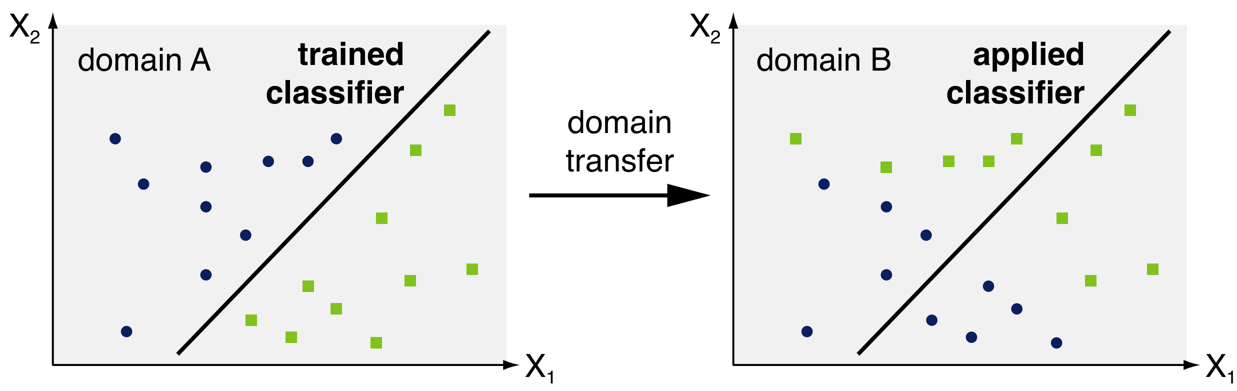




Domain dependency

- **Domain**
  - A set of texts that share certain properties
  - Can refer to a topic, genre, style, or similar — or combinations
  - Texts from the same domain often have a similar feature distribution.

- **Domain dependency**
  - Many algorithm work better in the domain of training texts than in others.

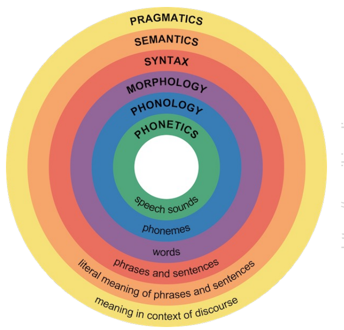


  - The same feature values result in different output information.
  - Different features are discriminative regarding the target variable.
    Example: ”Read the book” in book reviews vs. movie reviews... vs. hotel reviews?

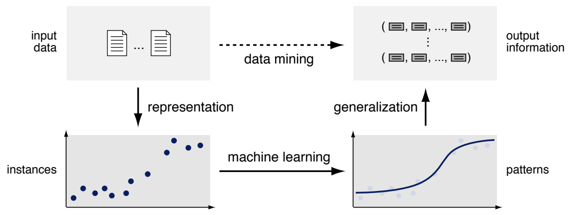
Next section: Conclusion

Conclusion

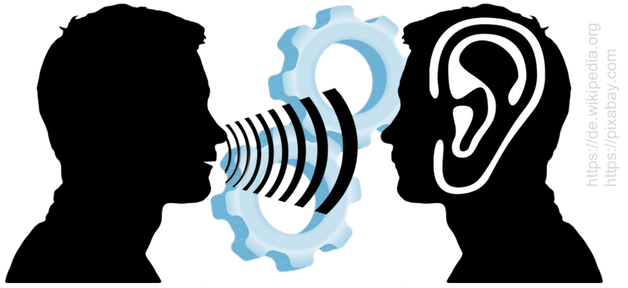
- **Basics of natural language processing (NLP)**
  - Linguistic knowledge from phonetics to pragmatics
  - Empirical methods for development and evaluation
  - Rule-based and statistical (machine-learned) algorithms



- **How to approach NLP tasks?**
  - Start from annotated text corpora
  - Develop algorithms that use rules or learn patterns
  - Evaluate quality of their output empirically



- **Goals of NLP**
  - Technology that can process natural language
  - Empirical explanations of linguistic phenomena
  - Solutions to problems from the real world



References

- **Ng (2018).** Andrew Ng. Machine Learning. Lecture slides from the Stanford Coursera course. 2018.https://www.coursera.org/learn/machine-learning.

- **Wachsmuth (2020).** Henning Wachsmuth. Introduction to Text Mining. Lecture slides. Winter term, 2020. https://cs.upb.de/css/teaching/courses/text-mining-w20/