

Statistical Natural Language Processing

Part III: Basics of Statistical NLP

Henning Wachsmuth

<https://ai.uni-hannover.de>

Learning Objectives

Concepts

- Corpus creation in 10 steps
- Fundamentals of machine learning
- The standard data mining process

Methods

- Computation of chance-corrected inter-annotator agreement
- Computation of correlation coefficients
- Standard techniques in machine learning
- Basic optimization procedures of machine learning models

Outline of the Course

I. Overview

II. Basics of NLP (video only)

III. Basics of Statistical NLP

- Introduction
- Corpus Creation
- Machine Learning
- Data Mining
- Conclusion

IV. Learning Representations

V. NLP using Clustering

VI. NLP using Classification and Regression

VII. NLP using Sequence Labeling

VIII. NLP using Neural Networks

IX. NLP using Transformers

X. Practical Issues

Introduction

Basics of Statistical NLP

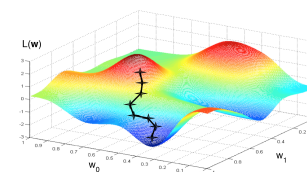
Corpus creation

- NLP methods are developed and evaluated on carefully designed collections of texts called *corpora*.
- Corpus creation is thus an important task in NLP.



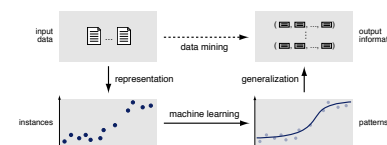
Machine learning

- NLP, as studied in this course, employs statistics using machine learning methods.
- Basics of machine learning are thus integral to NLP.



Data mining

- NLP analyzes (or synthesizes) data empirically to infer (or create) new or hidden information.
- In that, it follows the common process of data mining.



Corpus Creation

Text Corpora

Text corpus (plural text *corpora*)

- A principled collection of (mostly real-world) natural language texts with known properties, compiled to study a language problem

Examples: 200,000 product reviews for sentiment analysis,
1000 news articles for part-of-speech tagging, ...

- The texts in a corpus are often annotated, at least for the problem to be studied.

Examples: Sentiment polarity of a full text,
part-of-speech tags of each token, ...



<https://pixabay.com>

Dataset

- A subset of a corpus used for development or evaluation
- NLP methods are trained and tested on the datasets of a corpus.
- Without a corpus, it is hard to develop a strong method — and even harder to reliably evaluate it.

Corpus Creation in 10 Steps

Input

1. **Text compilation.** Choose the texts to be included.
2. **Annotation scheme.** Define for what variables to annotate the texts.
3. **Text preprocessing.** Prepare texts for annotation.

Annotation process

4. **Annotation sources.** Decide who provides annotations.
5. **Annotation guidelines.** Define how to annotate.
6. **Pilot annotation.** Test the annotation process.
7. **Inter-annotator agreement.** Compute how reliable the annotations are.

Output

8. **Postprocessing.** Fix errors and filter annotations.
9. **File representation.** Store the annotated texts adequately.
10. **Dataset splitting.** Create subsets for training and testing.

Corpus Creation in 10 Steps

1. Text Compilation

Text compilation

- The first step is to collect the texts to be included.
- The texts should represent the application scenario of the task studied.
- Several types of potential data bias may need to be accounted for.
- Also, copyrights have to be considered.

Main design decisions

- **Size.** Usually, the more the better, but annotation must remain doable
- **Domains.** Topics, genres, languages, etc. (or combinations) to consider
- **Confounders.** Variables to control for (via balancing, defined ranges, ...)

Examples: Publication time, length, or author

Example: ArguAna TripAdvisor corpus (Wachsmuth et al., 2014)

- 2100 English hotel reviews to annotate (+ 196k extra)
- 300 reviews each for 7 locations, 420 each with rating 1–5

All reviews taken from an existing corpus (Wang et al., 2010)



<https://pixabay.com>

Corpus Creation in 10 Steps

1. Text Compilation: Representativeness and Balance

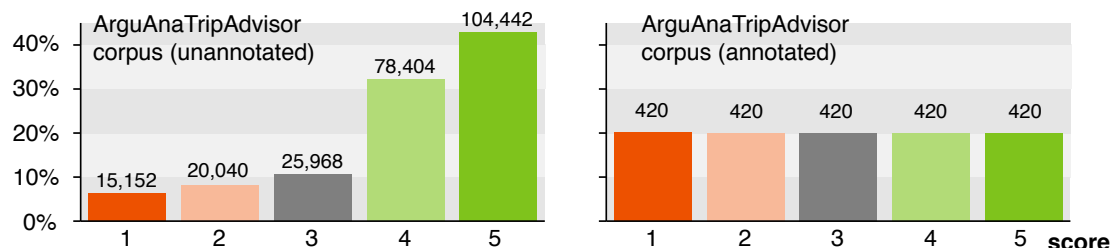
Representativeness

- A text compilation is representative for some variable X , if it includes the full range of variability with respect to X .
- What texts to include governs what can be learned about a given task.
- For evaluation, a representative distribution of texts is usually favorable.

Balance

- A text compilation is balanced if all values of X are represented evenly.
- For development, a balanced distribution may be favorable.

Example: ArguAna TripAdvisor corpus



Corpus Creation in 10 Steps

2. Annotation Scheme

Annotation scheme

- The definition of all annotation types to be considered
- The scheme clarifies syntax, semantics, and/or pragmatics of all types.
- It governs what can be studied for a task explicitly on a corpus.

Example: ArguAna TripAdvisor corpus

- **Sentiment.** Each statement classified as positive, negative, or neutral
A statement was defined to be ≥ 1 clause, ≤ 1 sentence, and meaningful on its own.
- **Aspects.** Each aspect of a hotel marked
- **Ratings.** Each review scored for several quality dimensions

title: *great location, bad service* **sentiment score:** 2 of 5

body: *stayed at the darling harbour holiday inn. The location was great, right there at China town, restaurants everywhere, the monorail station is also nearby. Paddy's market is like 2 mins walk. Rooms were however very small. We were given the 1st floor rooms, and we were right under the monorail track, however noise was not a problem. Service is terrible. Staffs at the front desk were impatient. I made an enquiry about internet access from the room and the person on the phone was rude and unhelpful. Very shocking and unpleasant encounter.*

Corpus Creation in 10 Steps

3. Text Preprocessing

Text preprocessing

- The preparation of corpus texts for their manual annotation

Usual preprocessing steps

- Input files are converted into a common, usually simple format.
- Metadata is stored, in case it is considered relevant.
- The instances to be annotated are derived from the corpus texts.

Example: ArguAna TripAdvisor corpus

- Originally, the input reviews were crawled HTML pages.
- The review contents were converted to plain text.
- Ratings and other metadata were stored in annotations.
- Each text was pre-segmented automatically into statements.

The rule-based segmentation algorithm used is provided with the corpus.

Corpus Creation in 10 Steps

4. Annotation Sources

Expert annotation

- Experts for a task or domain manually annotate each corpus text.
- Usually best results, but often time and cost intensive

Crowd-based annotation

- A crowdsourcing platform is used for manual annotation
- Access to many lay annotators (cheap) or semi-experts (not that cheap)
- Distant coordination overhead; results for complex tasks less reliable

Distant supervision

- Annotations are (semi-)automatically derived from existing metadata.
- Enables large corpora, but annotations may be noisy

Example: ArguAna TripAdvisor corpus

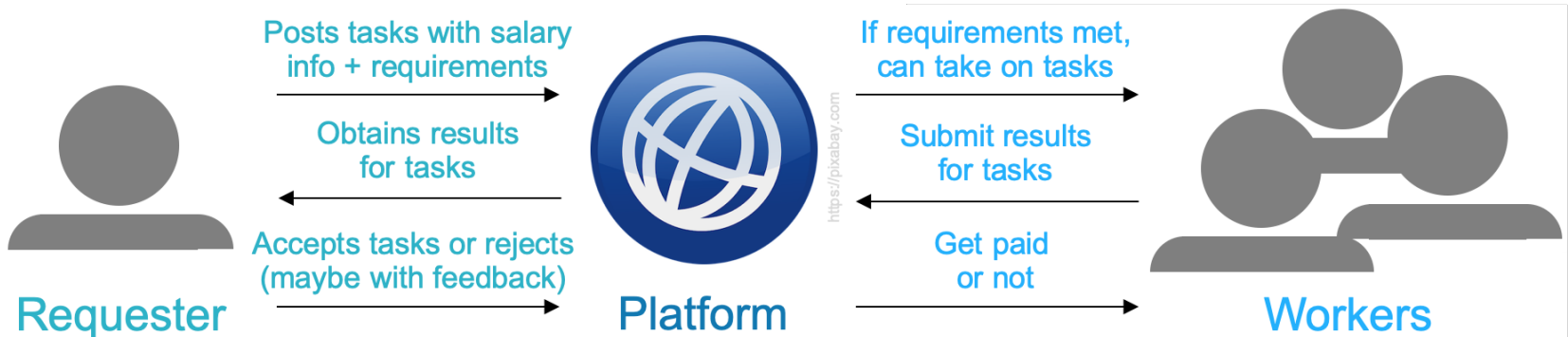
- **Sentiment.** Crowd-based annotation, three annotators each
- **Aspects.** Expert annotations, one expert per review (two for a sample)
- **Ratings.** Distant supervision; ratings obtained from review metadata

Corpus Creation in 10 Steps

4. Annotation Sources: Crowdsourcing

Crowdsourcing

- Outsourcing of (usually micro) jobs to people around the world
- Tasks and results are submitted to a crowdworking platform.



Selected platforms

- Amazon Mechanical Turk (AMT). Biggest platform, lay workers
- Upwork. Semi-professional freelancers for several areas

Example: ArguAna TripAdvisor corpus

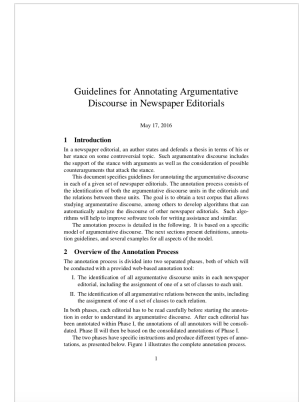
- AMT, \$0.05 per 12 sentiment classifications, 328 workers involved

Corpus Creation in 10 Steps

5. Annotation Guidelines

Annotation guidelines

- To obtain reliable annotations, the annotators get guidelines on what and how to annotate.
- Guidelines include concepts, the annotation scheme, the annotation process, and often examples.



Length as a design decision

- Guidelines may span dozens of pages, but may also be very short.
- The more complete, the more annotations will reflect the authors' view.
- The more concise, the more decisions are left to the annotators' view.

Example: ArguAna TripAdvisor corpus

- For crowd-based sentiment, we had very short guidelines (+ examples):

“When visiting a hotel, are the following statements positive, negative, or neither?”

Notes. (1) Pick *neither* only for facts, not for unclear cases. (2) Pay attention to subtle statements where sentiment is expressed implicitly or ironically. (3) Pick the most appropriate answer in controversial cases.

Corpus Creation in 10 Steps

6. Pilot Annotation

Pilot annotation

- Before a complete corpus is annotated, guidelines are usually tested on a small sample.
- Goal: Identify unclear parts, overseen and hard cases, and general annotation problems.
- Guidelines are often written incrementally based on pilot studies.



<https://pixabay.com>

Annotators in pilot study

- Experts may discuss and align their annotation based on pilot results.
- Sometimes, a subset of annotators is filtered based on pilot results.
- **Rule of thumb.** If authors don't agree, annotators won't either.

Al Khatib et al. (2016) omitted to annotate argumentative relations for this reason.

Example: ArguAna TripAdvisor corpus

- **Sentiment.** The guideline above was best among multiple variations.
- **Aspects.** The decision to use experts was based on pilot crowd tests.

Corpus Creation in 10 Steps

7. Inter-Annotator Agreement

Inter-annotator agreement (IAA) (aka inter-rater reliability, inter-coder agreement)

- Quantification of the similarity of annotations by multiple annotators
- From 1.0 (perfect agreement) to -1.0 (systematic disagreement)
- Often 2–5 annotators, sometimes more

Why inter-annotator agreement?

- Captures the reliability (or homogeneity) of the annotations of a corpus
- Gives a rough idea of how effective an algorithm may become
- **Dilemma.** Low agreement may indicate bad guidelines or insufficient training — but also just a subjective task.

Alternatives for computing agreement

1. Each corpus instance is annotated by multiple annotators.
2. A sample is annotated multiple times, and the rest once each.
#1 is statistically more reliable and allows filtering, majority voting, etc.; #2 is cheaper.

Corpus Creation in 10 Steps

7. Inter-Annotator Agreement: Overview of Measures

Joint probability measures

- **Observed agreement.** % of nominal instances where 2 annotators agreed
- **Full agreement.** % of instances where $k \geq 3$ annotators all agreed
- **Majority agreement.** % of instances where $> 50\%$ annotators agreed

Chance-corrected measures

- More robust, taking into account that agreement may be due to chance
- **Cohen's κ .** Difference of observed to chance agreement (see below)
- **Fleiss' κ .** “Generalization” of Cohen's κ to $k \geq 3$ annotators
- **Krippendorff's α .** Focus on disagreement cases, any k , any scale

Correlation measures

- Quantify the pairwise correlation of annotators for ordinal scale
- **Kendall's τ .** Concordance of ranks of two instance orderings (see below)
- **Spearman's ρ .** Monotonicity of the relation between two orderings
- **Pearson's r .** Linear correlation between two sets of continuous values

Corpus Creation in 10 Steps

7. Inter-annotator Agreement: Cohen's κ

Cohen's κ

- For n instances annotated by annotators A and B and a set of nominal classes C :

$$\kappa := \frac{p_o - p_e}{1 - p_e} \quad \text{where} \quad p_e := \frac{1}{n^2} \sum_{c \in C} a_c \cdot b_c$$

- p_o : Observed agreement on instances
- p_e : Expected agreement by chance
- a_c, b_c : Number of times A and B chose class c

rough interpretation

κ Range	Agreement
[-1.0, 0.0]	No
(0.0, 0.2]	Slight
(0.2, 0.4]	Fair
(0.4, 0.6]	Moderate
(0.6, 0.8]	Substantial
(0.8, 1.0]	"Perfect"

Example

- $n = 100$, $p_o = 0.75$ for two categories c and c' ($a_c = b_c = 80$, $a_{c'} = b_{c'} = 20$)

$$p_e = \frac{1}{100^2} \cdot (6400 + 400) = 0.68 \quad \text{and thus} \quad \kappa = \frac{0.75 - 0.68}{1 - 0.68} \approx 0.22$$

Example: ArguAna TripAdvisor corpus

- Sentiment.** Fleiss' $\kappa = 0.67$, full 73.6%, majority 98.3%
- Hotel aspects.** Cohen's $\kappa = 0.73$ (based on 546 cases)

Corpus Creation in 10 Steps

7. Inter-annotator Agreement: Kendall's τ

Kendall's τ rank correlation coefficient

- Given n instances to be ranked, let $(a_1, b_1), \dots, (a_n, b_n)$ be their joint ranks assigned by annotators A and B . Then:

$$\tau := \frac{\#\text{concordant pairs} - \#\text{discordant pairs}}{n \cdot (n - 1)/2}$$

- Concordant.** Any $(a_i, b_i), (a_j, b_j), i < j : a_i < a_j \ \& \ b_i < b_j$ or $a_i > a_j \ \& \ b_i > b_j$
- Discordant.** Any $(a_i, b_i), (a_j, b_j), i < j : a_i < a_j \ \& \ b_i > b_j$ or $a_i > a_j \ \& \ b_i < b_j$

Adjustment for ties

- τ ignores the number of ties, t_A (for $a_i = a_j$) and t_B (for $b_i = b_j$).
- A common adjustment, τ' , replaces the denominator of τ by:

$$\sqrt{(\#\text{conc.} + \#\text{disc.} + t_A) \cdot (\#\text{conc.} + \#\text{disc.} + t_B)}$$

Example

- $n = 3$, rank pairs: $(1, 2), (2, 3), (3, 3)$
- $\#\text{conc.} = 2, \#\text{disc.} = 0, t_A = 0, t_B = 1$

$$\tau = (2 - 0)/3 \approx 0.67$$

$$\tau' = (2 - 0)/\sqrt{6} \approx 0.82$$

Corpus Creation in 10 Steps

8. Postprocessing

Postprocessing

- The consolidation of the annotated texts for the final corpus
- Includes *cleansing* of potentially wrong or inconsistent cases
- May be manual and/or automatic

Common postprocessing steps

- Resolution (or discarding) of cases where annotators disagreed
- Removal of noise in the data observed during annotation
- Merging of labels that have been assigned only rarely

Example: ArguAna TripAdvisor corpus

- Each statement was assigned its majority sentiment where available.
- The 1.7% sentiment disagreement cases were resolved manually in the context of their associated reviews.
- Wrong aspect annotation boundaries were fixed automatically.

Corpus Creation in 10 Steps

9. File Representation

File representation

- Usually, each corpus text is stored in a separated file.
- Large corpora may be stored in databases or indexes.
- Various file representations exist.



Common file representations

- **Text only.** One line per token, one tab per token-level annotation
- **Text + annotation.** Plain text, extra file (e.g., JSON) for annotations
- **XMI/XML.** One file for each text, one tag per annotation
- **Spreadsheet.** One row per text, one additional column per annotation

Example: ArguAna TripAdvisor corpus

- XMI files preformatted for the Apache UIMA framework
- Each annotation is stored as a tag with attributes and character indices.
- The annotation scheme is specified in a global XMI file.

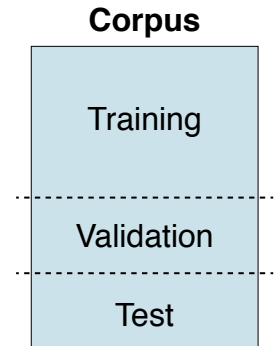
Corpus Creation in 10 Steps

10. Dataset Splitting

Dataset splitting

- How to split a corpus into training, validation, and test set (or similar), depends on the task.
- Goal: Mimic the real-world situation of interest, and avoid *data leakage* that may be exploited in learning.

Example: Annotations of one text should usually be in different sets.



Common splitting criteria

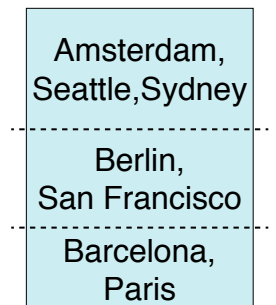
- **Random.** Split done (pseudo-) randomly
- **Topic.** Datasets (more or less) disjunct in terms of topic
- **Time.** Oldest texts for training, newest for testing

Often, good splitting criteria are task or dataset-specific.

Example: ArguAna TripAdvisor corpus

- **Location.** 3 locations for training, 2 for validation, 2 for test

This way, location-specific sentiment indicators cannot be exploited.



Machine Learning

Machine Learning

Example: Decision Making



Learning task

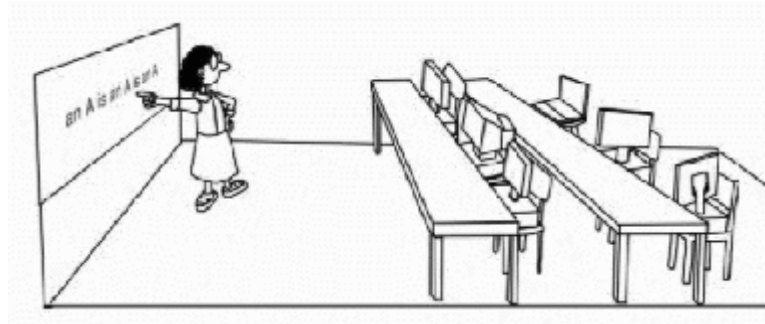
- What criteria form the basis of a decision?
- How is the decision made?

Machine Learning

Definitions

Machine learning (Samuel, 1959)

- The ability of an algorithm to learn without being explicitly programmed



An algorithm is said to learn... (Mitchell, 1997)

- ... from **experience**
- ... with respect to a given prediction **task**
- ... and some **performance** measure,
- ... if its **performance** on the **task** increases with the **experience**.

Machine Learning

Prediction

Prediction task

- A real-world problem that can be solved by a *target function* $\gamma : O \rightarrow C$
- **Input.** Objects o_1, o_2, \dots of some real-world concept O
- **Output.** Information c_1, c_2, \dots of some target variable C

The values of C are all of the same kind, for instance, all nominal labels.

(Ideal) Target function γ

- A function that interprets any object $o \in O$ to infer $\gamma(o) \in C$
- γ is operationalized by a human or some other real-world mechanism.
- Machine learning aims at prediction tasks where γ is unknown.

This includes most NLP tasks, also those that can be tackled well with rules.

Prediction using machine learning

- Machine learning finds statistical patterns in representations of objects from O that are relevant to infer information from C .

Machine Learning

Relation of NLP and Machine Learning

Machine learning in NLP

- **Task.** Predict output information $c \in C$ for a given text (or span of text).
- **Experience.** Training texts, annotated for C
- **Performance.** In terms of some effectiveness measure

Output information C

- **Text labels and scores.** Topic, sentiment, grades, ...
- **Span annotations.** Tokens, entities, ...
- **Span classifications.** Part-of-speech tags, entity types, ...
- **Span relations.** Entity relations, coherence relations, ...
- **Probabilities.** For example, of next words to generate

Two-way relationship

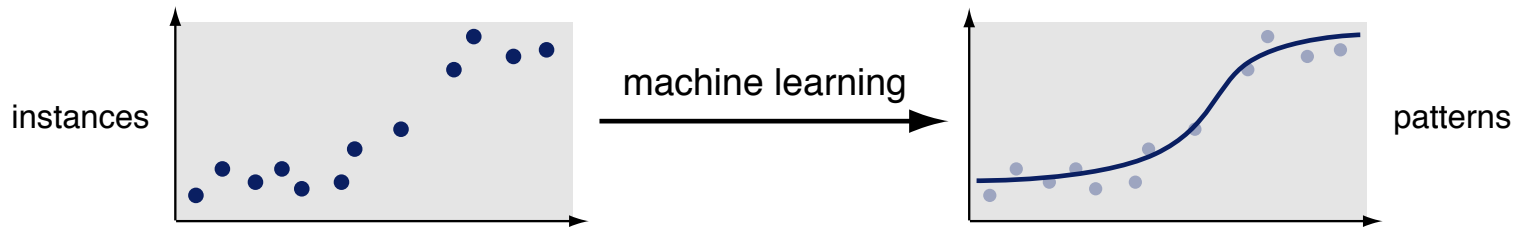
- NLP often uses machine learning to predict output information.
- Its output may be the input to machine learning, e.g., to train a classifier.

Machine Learning

Model

Machine learning models

- A model $y : X \rightarrow C$ is a mapping from formalized object representations X (called *instances*) to the target variable C
- y generalizes patterns found in X to approximate the target function γ .
Machine learning seeks for the optimal y with respect to some performance measure.



Model vs. target function

- γ and y differ in the complexity and representation of their domain.
- **Complexity.** Objects $o \in O$ are abstracted into (vector) instances $\mathbf{x} \in X$ using some representation function α , $\mathbf{x} = \alpha(o)$.
- **Representation.** $y(\mathbf{x})$ is the formalized counterpart of $\gamma(o)$.

Machine Learning

From the Real World to the Model

Real-world domain

- O is a real-world concept (or: the set of all objects), C a target variable.
- $\gamma : O \rightarrow C$ is the ideal target function for O .
- **Task.** Given some object $o \in O$, predict information $\gamma(o) \in C$.

Model domain

- X is the representation space (the set of all instances), C as before.
- $c : X \rightarrow C$ is the ideal predictor for X .
- **Task.** Given some instance $\mathbf{x} \in X$, predict information $c(\mathbf{x}) \in C$.

Example: Spam classification

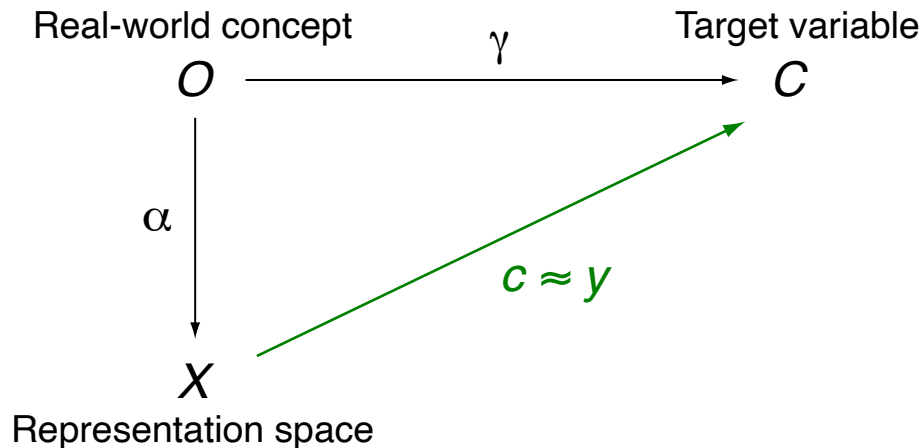
- O is the set of all emails, $C = \{\text{"spam"}, \text{"no spam"}\}$
- X may be the distribution of words in an email
- γ is a human expert on spam, c is a classifier
- **Task.** Given an email, is it spam or not?



<https://datenschutz.org>

Machine Learning

Overview of the Concepts



Notation

- γ Unknown ideal target function, mapping objects o to information c
- α Representation function
- c Unknown ideal predictor, mapping instances x to information c
- y Machine learning model to be learned
- $c \approx y$ c is approximated by y (based on a set of instances)

Machine Learning

How to Learn

Learning types

- Machine learning differs in terms of what kind of patterns are learned as well as to what kind of data it is applied to.
- **Major types.** *Supervised* and *unsupervised* learning
A third type is reinforcement learning, which we shortly cover much later in the course.

Major types in a nutshell

- **Supervised.** Derive a model from patterns in annotated training data (where the ground truth is known).
- **Unsupervised.** Derive model from unannotated data (no ground truth).

Learning algorithms

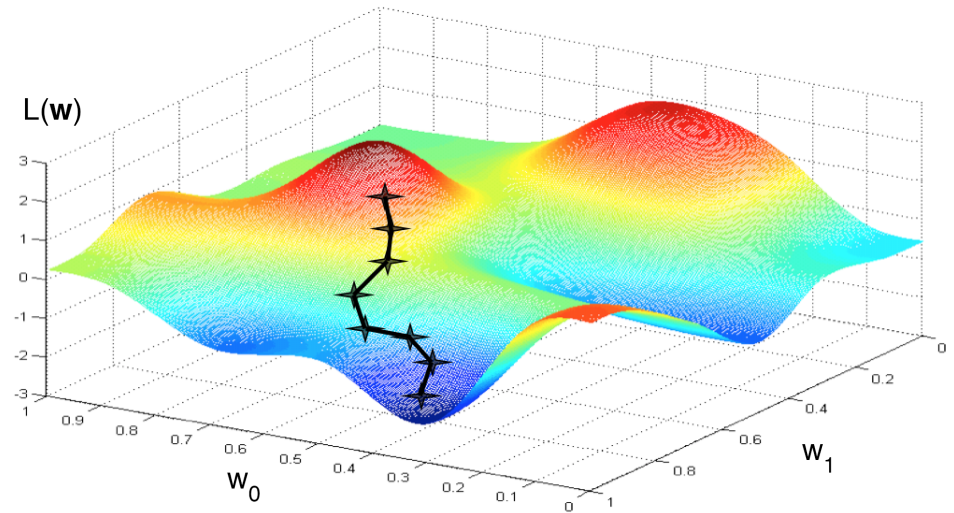
- Algorithms differ in terms of what patterns can be found, how they are represented, and how models are optimized.
- Some algorithms are discussed in detail in later lecture parts.

Machine Learning

Optimization

Training

- A learning algorithm incrementally creates candidate models y .
- y defines weights w for processing vectors x .
Not all learning algorithms assign weights explicitly.
- On a training set, y is tested against a *loss function* $\mathcal{L}(w)$.
 \mathcal{L} inversely reflects some performance measure.
- Based on $\mathcal{L}(w)$, w is adapted to create the next model y' .
- For this, an *optimization procedure* is used, such as gradient descent.



Hyperparameter optimization

- Most learning algorithms have *hyperparameters* whose best values depend on how well the training data reflects the real distribution.
- Hyperparameters need to be optimized against a validation set.

Supervised Learning

Supervised (machine) learning

- A learning algorithm builds a model y on *known* training data, i.e., pairs of a vector $\mathbf{x}^{(i)}$ and correct output information $c(\mathbf{x}^{(i)})$.
- y can then be used to predict output information for unknown data.

Why “supervised”?

- The learning process is guided by instances of correct predictions.



Supervised classification vs. regression

- **Classification.** Assign a nominal class to an instance.
- **Regression.** Predict a numeric value for an instance.

Manifold applications in NLP

- **Classification.** Standard technique for any text classification task, for extracting relations between entities, and similar
- **Regression.** Used to predict scores, ratings, probabilities, ...

Supervised Learning

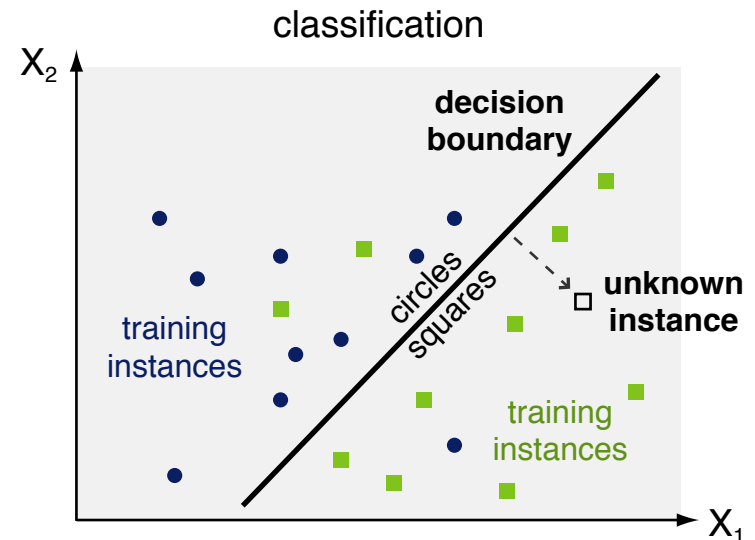
Classification

Classification

- The task to assign an object to the most likely of a set of two or more predefined discrete classes

Supervised classification

- An optimal decision boundary y is sought for on training vectors X with known classes C .
- The boundary decides the class of unknown instances.



Binary vs. multiple-class classification

- Binary classifiers separate the instances of two classes.
- Multiple classes are handled through multiple binary classifiers, e.g., using one-versus-all classification.

Supervised Learning

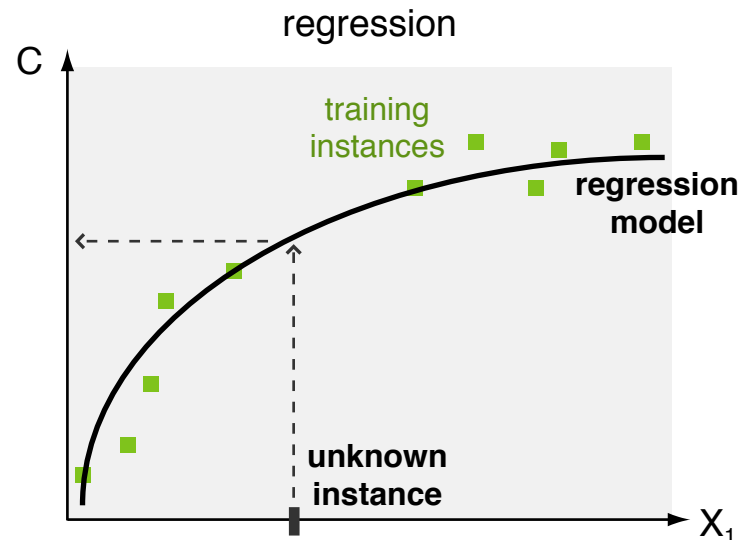
Regression

Regression

- The task to assign a given object to the most likely value of a real-valued, continuous target variable

Supervised regression

- An optimal regression function y is sought for on training vectors X with known values C .
- The function decides the value of unknown instances.



Value range in regression

- The values c predicted by y are not bounded, so $c \in (-\infty, \infty)$.
- Training on a defined value range does not fully prevent outliers.
Postprocessing may be needed to guarantee staying in the defined range.

Unsupervised Learning

Unsupervised (machine) learning

- A model y is derived from vectors X only, without output information.
- y reveals the organization and association of input data.
- **Techniques.** Clustering, autoencoders, principal component analysis, ...
The focus will be on clustering in this course.

Clustering

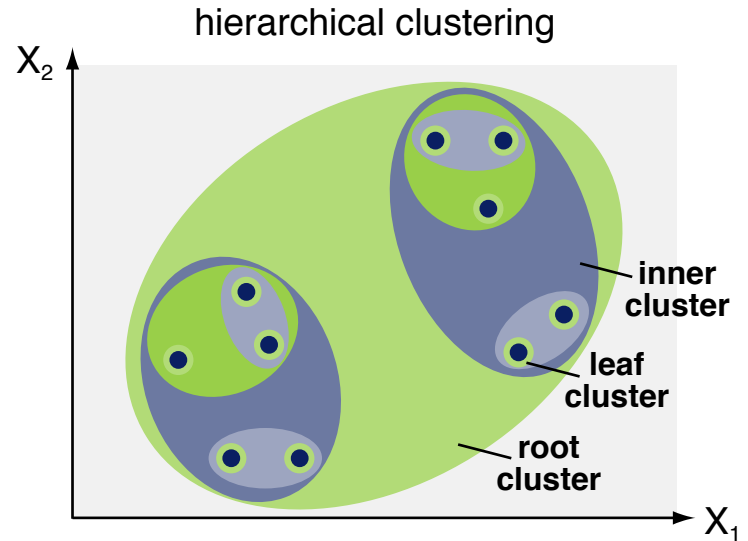
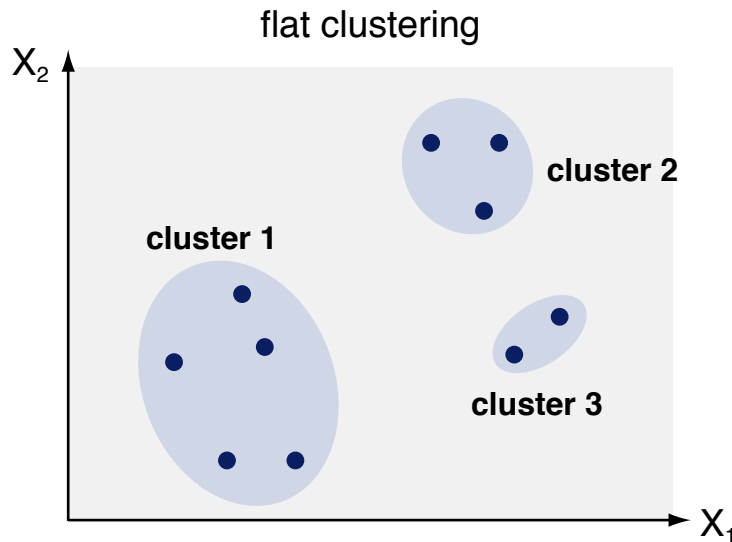
- The grouping of a set of instances into a possibly but not necessarily predefined number of classes (aka *clusters*).
The meaning of a class is usually unknown in advance.
- **Hard clustering.** Each instance belongs to a single cluster.
- **Soft clustering.** Instances belong to each cluster with some weight.

Applications in NLP

- Detection of texts with similar properties, mining of topics, ...

Unsupervised Learning

Flat vs. Hierarchical Clustering



Flat clustering

- Group instances into a (possibly predefined) number of clusters.
- No associations between the clusters are specified.

Hierarchical clustering

- Create a binary tree over all instances.
- Each tree node represents a cluster of a certain size.

Data Mining

Data Mining

Data mining

- The inference of new (or “hidden”) output information of specified types from typically huge amounts of input data
- Data mining deals with prediction tasks.

Data mining in a nutshell

- **Representation.** Map data to instances of a defined representation.
- **Machine learning.** Find statistical patterns in the instances that are relevant to the prediction task (aka *training*).
- **Generalization.** Apply the found patterns to infer new information from unseen data (aka *prediction*).

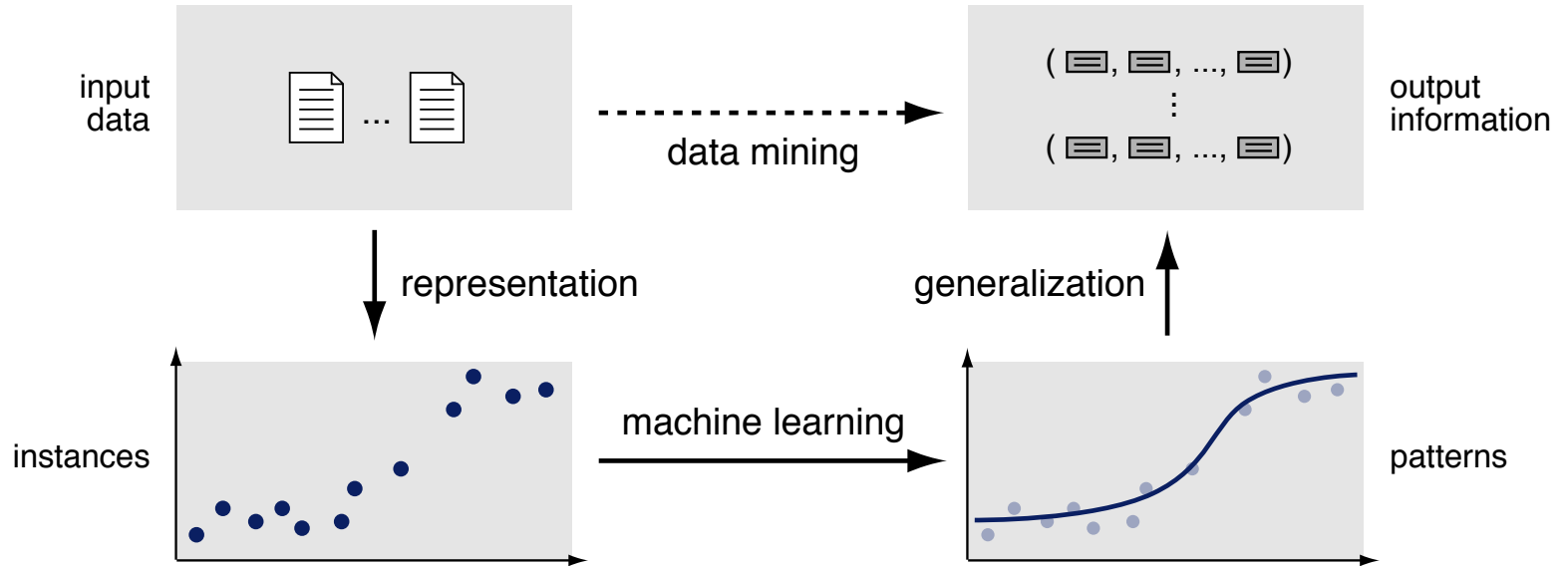
NLP as data mining

- **Input data.** A text corpus, i.e., a collection of texts to be processed
- **Output information.** Annotations of (spans of) the texts, or new texts
- The *representation* step is what makes NLP specific.

Data Mining

Process

The data mining process



Data Mining

Representation

Representation of instances

- Given a task to predict some type C , each object $o_i \in O$ is mapped to a common form.

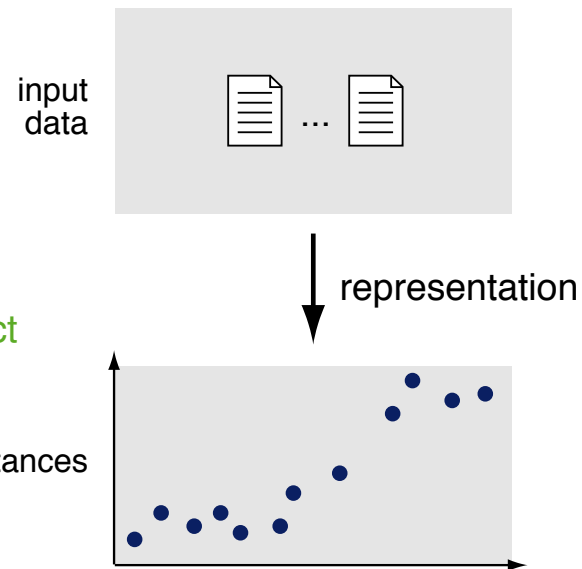
Sentiment analysis: each text (span) is one object
Entity recognition: each candidate entity is one object

Feature representations

- Map o_i to a (sparse) vector of values $\mathbf{x}^{(i)}$.
- This is done with a function $\alpha : O \rightarrow X$.
- What the features $x \in \mathbf{x}^{(i)}$ reflect is defined by the human developer.

Embedding representations (note: \mathbf{v} and V used to stress difference to above)

- Map o_i to one or more (dense) vectors of values $\mathbf{v}^{(i)}$.
- This is done with a function $\alpha : O \rightarrow V$.
- Embeddings $\mathbf{v} \in V$ are learned from distributional similarities of inputs.

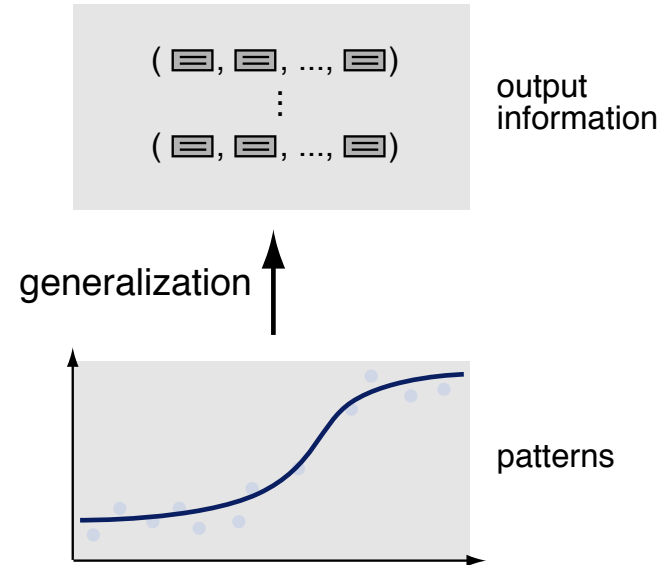


Data Mining

Generalization

Generalization

- Application of the learned model y to unseen data to infer new information.
- How well y generalizes depends on how well it fits the target function γ .
- Generalization is mainly decided by the training process (see above).



Bias in training

- The training process explores a large space of models $Y = \{y_0, y_1, \dots\}$.
- An important training decision is how much to bias the process wrt. the complexity of the model y to be learned.

Data Mining

Underfitting and Overfitting

Simple vs. complex models

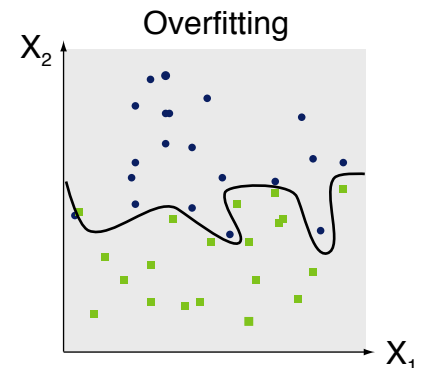
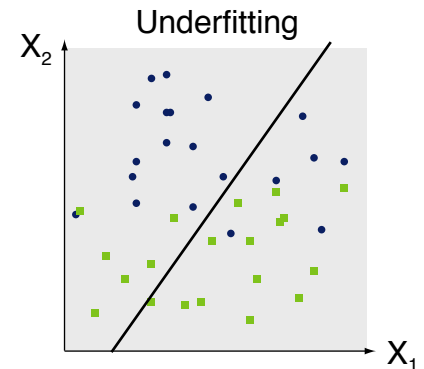
- **Simple.** Induce high bias to avoid noise; may underfit the input data
 - **Complex.** Induce low bias to fit the input data well; may capture noise
- Simple models may, e.g., be linear functions, complex models high polynomials.

Underfitting (too high bias)

- A model y generalizes too much, not capturing all relevant properties of the training data.
- y is too simple and will have limited effectiveness.

Overfitting (too high variance)

- A model y captures both relevant and irrelevant properties of the training data.
- y is too complex and will thus not generalize well.



Data Mining

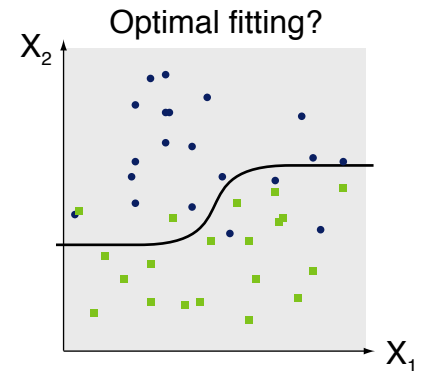
Optimal Fitting and Regularization

Avoiding underfitting and overfitting

- The best way to avoid both is to achieve an *optimal fitting*.
- Overfitting can also be countered through *regularization*.

Optimal fitting

- A model y perfectly approximates the complexity of γ based on the training data.
- In general, the right complexity is unknown.



Regularization

- Refrain from making y complex, unless it significantly reduces the loss.
- This is done by adding a term to the loss function that forces the feature weights to be small.

More on regularization in a later part of this course.

Conclusion

Conclusion

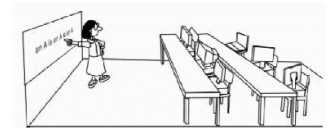
NLP and corpus creation

- Text corpora needed for development and evaluation
- Creation means to compile, annotate, and consolidate
- Key aspects include agreement and dataset splits



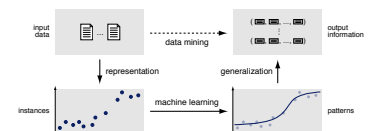
NLP and machine learning

- Approximating target functions in prediction tasks
- Inferring models from statistical patterns in training sets
- Focus here on supervised and unsupervised learning



NLP and data mining

- Inference of output information from huge input data
- Representation, machine learning, and generalization
- NLP can be seen as data mining on text



References

Some content taken from

- **Ng (2018)**. Andrew Ng. Machine Learning. Lecture slides from the Stanford Coursera course, 2018. <https://www.coursera.org/learn/machine-learning>
- **Stein and Lettmann (2010)**. Benno Stein and Theodor Lettmann. Machine Learning. Lecture Slides, 2010. <https://webis.de/lecturenotes/slides.html#machine-learning>
- **Wachsmuth (2015)**. Henning Wachsmuth. Text Analysis Pipelines — Towards Ad-hoc Large-scale Text Mining. LNCS 9383, Springer, 2015.
- **Witten and Frank (2005)**. Ian H. Witten and Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers, San Francisco, CA, 2nd edition, 2005.

References

Other references

- **Al Khatib et al. (2016)**. Khalid Al Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. A News Editorial Corpus for Mining Argumentation Strategies. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3433–3443, 2016.
- **Mitchell (1997)**. Tom M. Mitchell. Machine Learning. McGraw Hill, 1997.
- **Samuel (1959)**. Arthur Samuel. Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development, 44:206–226, 1959.
- **Wachsmuth et al. (2014)**. Henning Wachsmuth, Martin Trenkmann, Benno Stein, Gregor Engels, and Tsvetomira Palarkarska. A Review Corpus for Argumentation Analysis. In Proceedings of the of the 15th International Conference on Intelligent Text Processing and Computational Linguistics, pages 115–127, 2014.
- **Wang et al. (2010)**. Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach. In: Proceedings of the 16th SIGKDD. pages 783–792, 2010.