Machine Learning Basics

Learning Machine Learning

Nils Reiter

CENTER FOR REFLECTED TEXT ANALYTICS

September 26-27, 2018

Section 1

Text Analysis Experiments

Text Analysis Experiments

- Experiment
  - Reproducability
- Hypotheses about the operationalization of text phenomena
  - Linguistic: Syntax, Semantics, ...
  - Literary: Narratological categories (e.g., narrative levels), ...

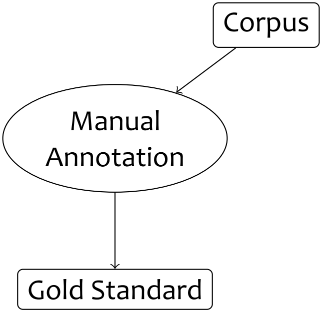
Example

- Position within a sentence is indicative for the part of speech
- Meaning of a word depends on its context

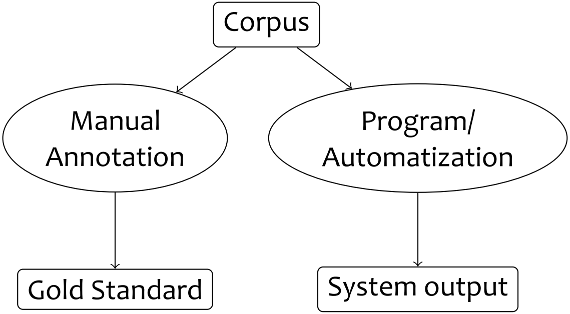
Text Analysis Experiments

Corpus

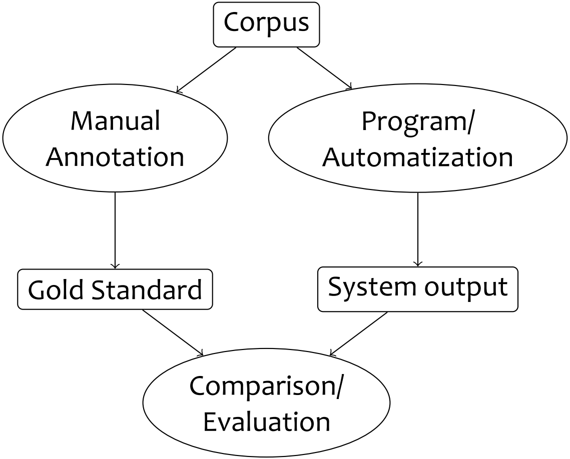
Text Analysis Experiments



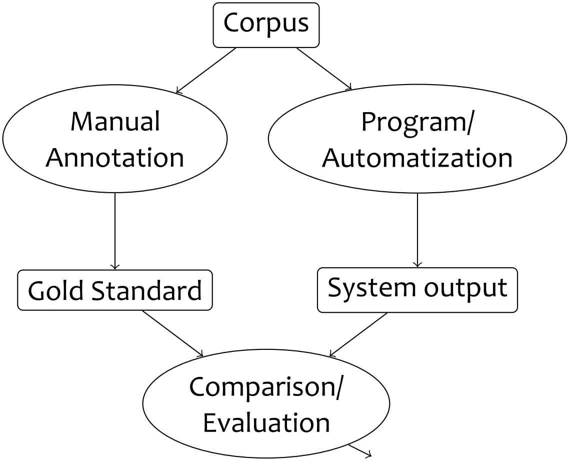
Text Analysis Experiments



Text Analysis Experiments

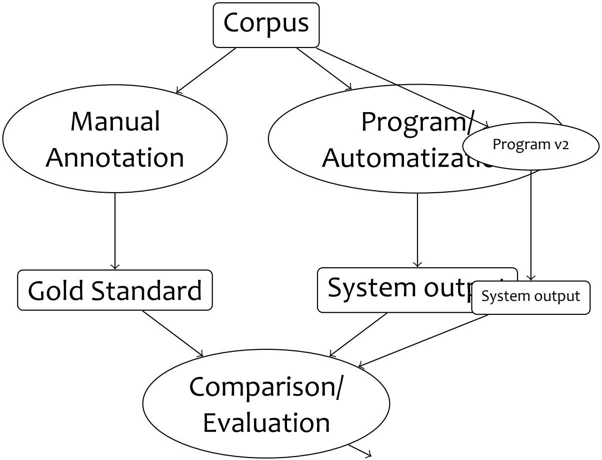


Text Analysis Experiments



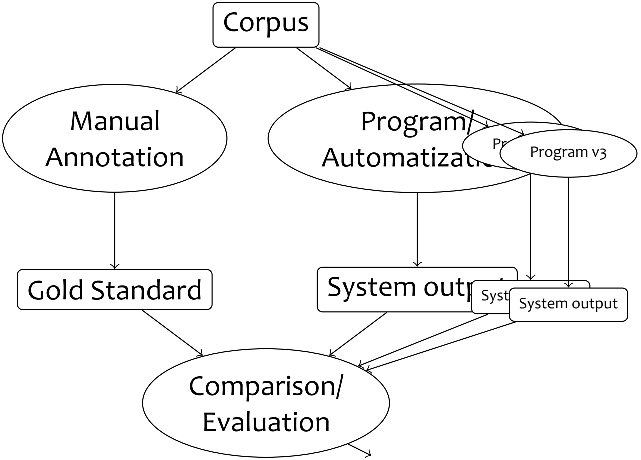
	P	R	F
Programm	5.7	9.2	...

Text Analysis Experiments



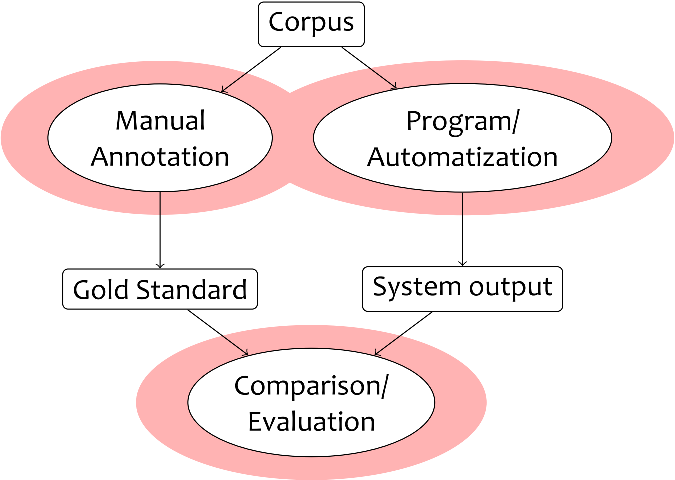
	P	R	F
Programm	5.7	9.2	...
v2	9.9	16.7	...

Text Analysis Experiments



	P	R	F
Programm	5.7	9.2	...
v2	9.9	16.7	...
v3	15.3	21.8	...

Text Analysis Experiments



What do we need?

- Gold standard
  - Formal, machine-readable truth
- Program, that implements a given algorithm (which operationalizes our hypotheses)
- Evaluation metric
  - Formalized comparison of annotations

What do we learn?

- Directly
  - Prediction quality of the program on this corpus
- Indirectly
  - Insights, why the program works well (or not)
  - Estimation of the quality on other corpora
- Long term
  - Iterative improvement of the programs (e.g., in shared tasks)

Three Areas

- Manual Annotation
  - Annotated corpora encode language intuitions of (native) speakers
  - Explicit/machine-readable encoding of text properties
  - Annotation guidelines describe categories and how to handle difficult cases
    - https://sharedtasksinthedh.github.io/2017/10/01/howto-annotation/
- Automatization (see below)
- Evaluation
  - Quantification of correctness
  - Accuracy: Portion of correctly labeled instances
  - Precision/Recall/F-Score: Insight into class imbalances

Section 2

Automatization

Systems

- Predicts annotations
- Ideally: The same annotations as a human (the correct ones)
- Parameters
  - On what exactly does the program make predictions?
  - What information, criteria and features does it need?

System types

- Rule-based
- Statistical
  - Supervised
  - Unsupervised

Rule-based Systems

- Manually specified rules over certain criteria
  - HPSG grammar, XML-Parsing
- Criteria: Vocabulary from which rules are created
  - Noun: Every token, that starts with an upper case letter
  - Noun: Every token, that starts with an upper case letter and is not sentence initial

Supervised Systems

- Learn probabilities from annotated data
  - POS tagger
- More exact: Probabilities, that features X are associated with category Y
  - $P(\text{Noun}|\text{Upper case})$
  - $P(\text{Noun}|\text{Upper case and not sentence initial})$

Unsupervised Systems

- Predictions over features without training data and defined categories
  - topic modeling — Blei et al. (2003)
  - clustering
- Advantage: No training data
- Disadvantage: Results often difficult to interpret

Mixed systems

- Rules that are weighted on training data
- Semi-supervised
  - Annotated und not annotated data
- Bootstrapping
  - Unsupervised methods to create training data, then supervised systems

Features

- Feature extraction
  - "Translation" of the corpus into feature vectors
- Feature engineering
  - Design and implementation of feature extractors
- Linguistic features need to be determined somehow
  $\rightarrow$ Dependencies, modularization
- Playground!

Example: Parts of Speech

Features	Data type
Case	Binary
Length	> 0

Table: Features

Token	Case	L.
Der	u	3
Hund	u	4
bellt	l	5
.	?	1
Die	u	3
Katze	u	5
schnurrt	l	8
.	?	1

Table: Feature extraction

Example: Parts of Speech

Feature	Data type
Case	Binary
Length	> 0
Sentence initial	Binary

Table: Features

Token	Case	L.	S. initial
Der	u	3	Y
Hund	u	4	N
bellt	l	5	N
.	Jein	?	N
Die	u	3	Y
Katze	u	5	N
schnurrt	l	8	N
.	?	1	N

Table: Feature extraction

Example: Parts of Speech

Feature	Data type
Case	Binary
Length	> 0
Sentence initial	Binary

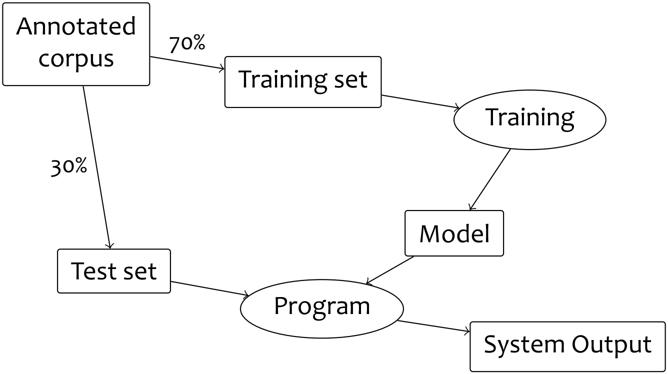
Introduces dependency!

Token	Case	L.	S. initial
Der	u	3	Y
Hund	u	4	N
bellt	l	5	N
.	Jein	?	N
Die	u	3	Y
Katze	u	5	N
schnurrt	l	8	N
.	?	1	N

Table: Feature extraction

Workflow

- Goal: Predict the quality on new data
- The program cannot have seen the data, so that it's a realistic test



Section 3

Text Analysis in the Digital Humanities

Text Analysis in the Digital Humanities

- Annotation workflow
  - Validation of theories (e.g., narratological)

Text Analysis in the Digital Humanities

- Annotation workflow
  - Validation of theories (e.g., narratological)
- Text processing/tools
  - Linguistic features for humanities phenomena

Text Analysis in the Digital Humanities

- Annotation workflow
  - Validation of theories (e.g., narratological)
- Text processing/tools
  - Linguistic features for humanities phenomena
- Automatic Annotation
  - “big data” investigations
    - e.g., all novels of the 19th century
  - Counteract canonization

Section 4

Machine Learning Concepts

Two Parts

Prediction Model

How do we make predictions on data instances?
(e.g., how do we assign a part of speech tag for a word?)

Learning Algorithm

How do we create a prediction model, given annotated data?
(e.g. how do we create rules for assigning a part of speech tag for a word?)

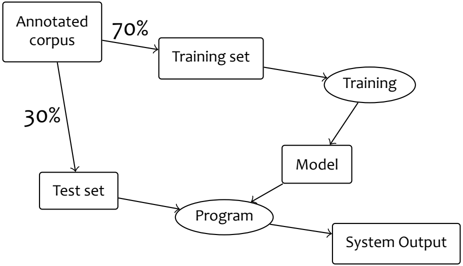
Two Parts

Prediction Model

How do we make predictions on data instances?
(e.g., how do we assign a part of speech tag for a word?)

Learning Algorithm

How do we create a prediction model, given annotated data?
(e.g. how do we create rules for assigning a part of speech tag for a word?)



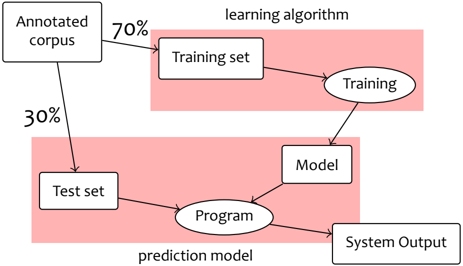
Two Parts

Prediction Model

How do we make predictions on data instances?
(e.g., how do we assign a part of speech tag for a word?)

Learning Algorithm

How do we create a prediction model, given annotated data?
(e.g. how do we create rules for assigning a part of speech tag for a word?)



Machine Learning

Classification

- Assigning *classes* to *objects/instances/items*

Machine Learning
Classification

- Assigning *classes* to *objects/instances/items*
  - Words $\rightarrow$ parts of speech

Machine Learning

Classification

- Assigning *classes* to *objects/instances/items*
  - Words $\rightarrow$ parts of speech
  - Photo portraits $\rightarrow$ gender of the depicted person

Machine Learning

Classification

- Assigning *classes* to *objects/instances/items*
  - Words $\rightarrow$ parts of speech
  - Photo portraits $\rightarrow$ gender of the depicted person
  - Photo portraits $\rightarrow$ name of depicted person

Machine Learning

Classification

- Assigning *classes* to *objects/instances/items*
  - Words $\rightarrow$ parts of speech
  - Photo portraits $\rightarrow$ gender of the depicted person
  - ~~Photo portraits $\rightarrow$ name of depicted person~~

Machine Learning
Classification

- Assigning *classes* to *objects*/*instances*/*items*
  - Words → parts of speech
  - Photo portraits → gender of the depicted person
  - ~~Photo portraits → name of depicted person~~
  - Texts → genres

Machine Learning

Classification

- Assigning *classes* to *objects/instances/items*
  - Words $\rightarrow$ parts of speech
  - Photo portraits $\rightarrow$ gender of the depicted person
  - ~~Photo portraits $\rightarrow$ name of depicted person~~
  - Texts $\rightarrow$ genres
- Prediction model: Responsible for the classification

Machine Learning

Classification

- Assigning *classes* to *objects/instances/items*
  - Words $\rightarrow$ parts of speech
  - Photo portraits $\rightarrow$ gender of the depicted person
  - ~~Photo portraits $\rightarrow$ name of depicted person~~
  - Texts $\rightarrow$ genres
- Prediction model: Responsible for the classification
- Many different models/algorithms available:
  - Decision trees
  - Support vector machines
  - Naïve bayes
  - Neural networks
  - Bayesian networks
  - ...

Machine Learning

Features

- Decision is based on features (= properties)
- The prediction model only sees feature values!
  - What's not encoded in a feature doesn't play a role
  - It's our job to provide useful features
    - …except when using neural networks: "deep learning"

Evaluation

- We *always* want to know how well machine learning works
- Straightforward evaluation: Comparison with a gold standard

Evaluation

- We *always* want to know how well machine learning works
- Straightforward evaluation: Comparison with a gold standard
- Most simple metric: Accuracy
  - Percentage of correctly classified instances (the higher the better)
  - Inverse: Error rate (percentage of incorrectly classified instances)

Evaluation

- We *always* want to know how well machine learning works
- Straightforward evaluation: Comparison with a gold standard
- Most simple metric: Accuracy
  - Percentage of correctly classified instances (the higher the better)
  - Inverse: Error rate (percentage of incorrectly classified instances)
- Accuracy is nice, but not enough
  - When improving systems, we want to *compare* our accuracy with the previous accuracy
  - When developing new systems, we want to know how difficult the task is
    - E.g., 60% accuracy when distinguishing 35 parts of speech is better than 60% accuracy when distinguishing nouns and all the rest

Evaluation

Baseline

Baseline

The baseline performance is the performance of a simple system, rule or thought experiment

Evaluation

Baseline

Baseline

The baseline performance is the performance of a simple system, rule or thought experiment

- Example 1: Gender of DH students
  - Task: Classify students according to their gender (Stuttgart DH class)
  - 22 of 25 students are female
  - Majority baseline: Everyone is female
  - Classification accuracy: 88% (!)

Evaluation

Baseline

Baseline

The baseline performance is the performance of a simple system, rule or thought experiment

- Example 1: Gender of DH students
- Example 2: Gender of arbitrary Germans
  - Task: Classify a random German according to their gender
  - male: 40.7m vs. female: 41.8m
  - Random baseline: Toss a coin
  - Classification accuracy: about 50%

Evaluation

Baseline

Baseline

The baseline performance is the performance of a simple system, rule or thought experiment

- Example 1: Gender of DH students
- Example 2: Gender of arbitrary Germans
- Example 3: Detecting nouns
  - Task: Classify words into noun and non-noun
  - Most words are not nouns
  - Majority baseline: Every word is a non-noun
  - Accuracy (in a German text): 81.8%

Evaluation

Typical baselines

Majority baseline

Always predict the majority class in the data set

Random baseline

Make a random selection

Single feature baseline

Make a prediction based on a single, easy to extract feature (e.g., casing of words)

Formalities and Notation

Why formal language?

Formal language is concise, exact and unambiguous. Slides will contain both.

Formalities and Notation

Why formal language?

Formal language is concise, exact and unambiguous. Slides will contain both.

- Data set D, split into D_{train} and D_{test}
  $D_{train} \cup D_{test} = D$

Formalities and Notation

Why formal language?

Formal language is concise, exact and unambiguous. Slides will contain both.

- Data set D, split into D_{train} and D_{test}
  $D_{train} \cup D_{test} = D$
- Data objects/instances/items: $x \in D$.
  x_{class} represents the class label (i.e., the target category)

Formalities and Notation

Why formal language?

Formal language is concise, exact and unambiguous. Slides will contain both.

- Data set D, split into D_{train} and D_{test}
  $D_{train} \cup D_{test} = D$
- Data objects/instances/items: $x \in D$.
  x_{class} represents the class label (i.e., the target category)
- Feature set $F = \{f_1, f_2, \ldots, f_n\}$
  - $v(f_i)$ is a set that contains all possible values of a feature
  - I.e., we know in advance which values a feature can take!

Formalities and Notation

Why formal language?

Formal language is concise, exact and unambiguous. Slides will contain both.

- Data set D, split into D_{train} and D_{test}
  $D_{train} \cup D_{test} = D$
- Data objects/instances/items: $x \in D$.
  x_{class} represents the class label (i.e., the target category)
- Feature set $F = \{f_1, f_2, \ldots, f_n\}$
  - $v(f_i)$ is a set that contains all possible values of a feature
  - I.e., we know in advance which values a feature can take!
- Feature extractor $f_i(x)$ represents the value of f_i for x

Formalities and Notation

Big operators

$$\sum_{\text{variable}} \text{expression}$$

Formalities and Notation

Big operators

$$\sum_{\text{variable}} \text{expression}$$

$\sum$ sum $\bigcup$ union $\max$ maximum $\arg$ argument

Formalities and Notation

Big operators

$$\sum_{\text{variable}} \text{expression}$$

$\sum$ sum $\bigcup$ union $\max$ maximum $\arg$ argument

$$\sum_{i \in \{1,2,3\}} i^2 = 1^2 + 2^2 + 3 + 2 = 14$$

Formalities and Notation

Big operators

$$\sum_{\text{variable}} \text{expression}$$

$\sum$ sum $\bigcup$ union $\max$ maximum $\arg$ argument

$$\sum_{i\in\{1,2,3\}} i^2 = 1^2 + 2^2 + 3 + 2 = 14$$

$$\max_{i\in\{1,2,3\}} i^2 = 9$$

Formalities and Notation

Big operators

$$\sum_{\text{variable}} \text{expression}$$

$\sum$ sum $\quad \bigcup$ union $\quad \max$ maximum $\quad \arg$ argument

$$\sum_{i \in \{1,2,3\}} i^2 = 1^2 + 2^2 + 3 + 2 = 14$$

$$\max_{i \in \{1,2,3\}} i^2 = 9$$

$$\underset{i \in \{1,2,3\}}{\operatorname{argmax}} i^2 = 3 \quad \text{(which } i \text{ leads to the maximum value?)}$$

References I

Blei, David, Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation". In: *Journal of Machine Learning Research* 3 (2003), pp. 993–1022.