Machine Learning Basics
Learning Machine Learning

Nils Reiter

C RETA Centre for Reflected Text Analytics

September 26-27, 2018
Text Analysis Experiments

Automatization

Text Analysis in the Digital Humanities

Machine Learning Concepts
  Classification
  Evaluation

Formalities and Notation
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

- Corpus
- Manual Annotation
- Gold Standard

Program/Programm 5.7 9.2 …

Program v2
Program v3
System output
Comparison/Evaluation

Nils Reiter (CRETA)
Machine Learning Basics
September 26-27, 2018
Text Analysis Experiments

Corpus

Manual Annotation
Gold Standard

Program/Automatization
System output

Comparison/Evaluation

P R F
Program 5.7 9.2 …
Program v2
System output
Program v3
System output

Nils Reiter (CRETA)
Text Analysis Experiments

- Corpus
  - Manual Annotation
  - Program/Automatization
  - Gold Standard
  - System output
- Comparison/Evaluation
Text Analysis Experiments

Corpus

Manual Annotation

Gold Standard

Program/Automatization

System output

Comparison/Evaluation

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programm</td>
<td>5.7</td>
<td>9.2</td>
<td>...</td>
</tr>
</tbody>
</table>
Text Analysis Experiments

Corpus

Manual Annotation

Gold Standard

Program/Automatization

Program v2

System output

Comparison/Evaluation

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programm v2</td>
<td>5.7</td>
<td>9.2</td>
<td>...</td>
</tr>
<tr>
<td>v2</td>
<td>9.9</td>
<td>16.7</td>
<td>...</td>
</tr>
</tbody>
</table>
Text Analysis Experiments

Corpus

- Manual Annotation
- Gold Standard

Program/Automatization

System output

Comparison/Evaluation

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program v2</td>
<td>5.7</td>
<td>9.2</td>
<td>...</td>
</tr>
<tr>
<td>v2</td>
<td>9.9</td>
<td>16.7</td>
<td>...</td>
</tr>
<tr>
<td>v3</td>
<td>15.3</td>
<td>21.8</td>
<td>...</td>
</tr>
</tbody>
</table>

Nils Reiter (CRETA)
Text Analysis Experiments

Corpus

Manual Annotation

Program/ Automatization

Gold Standard

System output

Comparison/ Evaluation

P R F

Program 5.7 9.2 …

Program v2

System output

Program v3

Nils Reiter (CRETA)

Machine Learning Basics

September 26-27, 2018
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
  - clustering

- Advantage: No training data
- Disadvantage: Results often difficult to interpret
Mixed systems

- Rules that are weighted on training data
- Semi-supervised
  - Annotated and 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
  → Dependencies, modularization
- Playground!
### Example: Parts of Speech

<table>
<thead>
<tr>
<th>Features</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case</td>
<td>Binary</td>
</tr>
<tr>
<td>Length</td>
<td>&gt; 0</td>
</tr>
</tbody>
</table>

**Table: Features**

<table>
<thead>
<tr>
<th>Token</th>
<th>Case</th>
<th>L.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Der</td>
<td>u</td>
<td>3</td>
</tr>
<tr>
<td>Hund</td>
<td>u</td>
<td>4</td>
</tr>
<tr>
<td>bellt</td>
<td>l</td>
<td>5</td>
</tr>
<tr>
<td>.</td>
<td>?</td>
<td>1</td>
</tr>
<tr>
<td>Die</td>
<td>u</td>
<td>3</td>
</tr>
<tr>
<td>Katze</td>
<td>u</td>
<td>5</td>
</tr>
<tr>
<td>schnurrt</td>
<td>l</td>
<td>8</td>
</tr>
<tr>
<td>.</td>
<td>?</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table: Feature extraction**
Example: Parts of Speech

<table>
<thead>
<tr>
<th>Feature</th>
<th>Data type</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Case</td>
<td>Binary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>&gt; 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentence initial</td>
<td>Binary</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table: Features**

<table>
<thead>
<tr>
<th>Token</th>
<th>Case</th>
<th>L.</th>
<th>S. initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Der</td>
<td>u</td>
<td>3</td>
<td>Y</td>
</tr>
<tr>
<td>Hund</td>
<td>u</td>
<td>4</td>
<td>N</td>
</tr>
<tr>
<td>bellt</td>
<td>l</td>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>Jein</td>
<td>?</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>Die</td>
<td>u</td>
<td>3</td>
<td>Y</td>
</tr>
<tr>
<td>Katze</td>
<td>u</td>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>schnurr</td>
<td>l</td>
<td>8</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>?</td>
<td>1</td>
<td>N</td>
</tr>
</tbody>
</table>

**Table: Feature extraction**
Example: Parts of Speech

<table>
<thead>
<tr>
<th>Feature</th>
<th>Data type</th>
<th>Feature</th>
<th>Data type</th>
<th>Feature</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case</td>
<td>Binary</td>
<td>Length</td>
<td>&gt; 0</td>
<td>Sentence initial</td>
<td>Binary</td>
</tr>
</tbody>
</table>

Table: Feature extraction

<table>
<thead>
<tr>
<th>Token</th>
<th>Case</th>
<th>L.</th>
<th>S. initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Der</td>
<td>u</td>
<td>3</td>
<td>Y</td>
</tr>
<tr>
<td>Hund</td>
<td>u</td>
<td>4</td>
<td>N</td>
</tr>
<tr>
<td>bellt</td>
<td>l</td>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jein</td>
<td>?</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>Die</td>
<td>u</td>
<td>3</td>
<td>Y</td>
</tr>
<tr>
<td>Katze</td>
<td>u</td>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>schnurrt</td>
<td>l</td>
<td>8</td>
<td>N</td>
</tr>
<tr>
<td>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table: Feature extraction
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 Concepts

Classification

Assigning classes to objects/instances/items
Machine Learning

Classification

- Assigning classes to objects/instances/items
  - Words → parts of speech
Machine Learning
Classification

▶ Assigning classes to objects/instances/items
  ▶ Words → parts of speech
  ▶ Photo portraits → gender of the 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
Assigning classes to objects/instances/items

- Words → parts of speech
- Photo portraits → gender of the depicted person
- Photo portraits → 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
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

Prediction model: Responsible for the classification
Machine Learning Concepts

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

- Prediction model: Responsible for the classification

- Many different models/algorithms available:
  - Decision trees
  - Support vector machines
  - Naïve bayes
  - Neural networks
  - Bayesian networks
  - ...

Nils Reiter (CRETA)
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
Baseline

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

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% (!)
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%
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 $D_{\text{train}}$ and $D_{\text{test}}$

$D_{\text{train}} \cup D_{\text{test}} = D$
Formalities and Notation

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

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

- Data objects/instances/items: $x \in D$.
  $x_{\text{class}}$ 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 $D_{\text{train}}$ and $D_{\text{test}}$
  \[ D_{\text{train}} \cup D_{\text{test}} = D \]
- Data objects/instances/items: $x \in D$.
  $x_{\text{class}}$ 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!
Why formal language?

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

- Data set $D$, split into $D_{\text{train}}$ and $D_{\text{test}}$
  \[D_{\text{train}} \cup D_{\text{test}} = D\]
- Data objects/instances/items: $x \in D$.
  $x_{\text{class}}$ 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

\[ \sum \text{ expression} \]

variable
Formalities and Notation

Big operators

\[ \sum \text{expression} \]

\[ \sum \text{sum} \quad \bigcup \text{union} \quad \text{max maximum} \quad \text{arg argument} \]
Formalities and Notation

Big operators

\[ \sum \text{ expression} \]
\[ \sum \text{ sum} \quad \bigcup \text{ union} \quad \max \text{ maximum} \quad \arg \text{ argument} \]
\[ \sum_{i \in \{1,2,3\}} i^2 = 1^2 + 2^2 + 3 + 2 = 14 \]
Formalities and Notation

Big operators

\[ \sum \text{expression} \]

\[ \sum \text{sum} \quad \bigcup \text{union} \quad \max \text{maximum} \quad \arg \text{argument} \]

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

\[ \max_{i \in \{1,2,3\}} i^2 = 9 \]
Formalities and Notation

Big operators

\[
\sum \text{ expression} \\
\sum \text{ sum} \quad \bigcup \text{ union} \quad \text{max} \text{ maximum} \quad \text{arg} \text{ argument}
\]

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

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

\[
\text{argmax}_{i \in \{1,2,3\}} i^2 = 3 \quad \text{(which } i \text{ leads to the maximum value?)}
\]