SEMINAR
KLASSIFIKATION & CLUSTERING

EVALUIERUNG VON KLASSIFIKATOREN

Stefan Langer

stefan.langer@cis.uni-muenchen.de

Evaluierung

- Ziel: Vergleich unterschiedlicher Klassifizierungsalgorithmen und Parametersetzungen, Identifizierung der geeignetsten Klassifizierungsmethode

- Erforderlich: vorklassifizierte Daten (Trainingskorpus, Testkorpus)

Korpora zur Evaluierung

- Ein Korpus ist eine Sammlung von Texten, u.U. verbunden mit Metadaten
- In der CL interessieren wir und in erster Linie für elektronische Korpora
- Für die Evaluierung von Textklassifikationsalgorithmen sind solche Korpora interessant, die bereits klassifiziert sind.
- Beispiele:
  - Zeitungskorpora mit Ressorts
  - Korpora bestimmter Texttypen

Our corpora

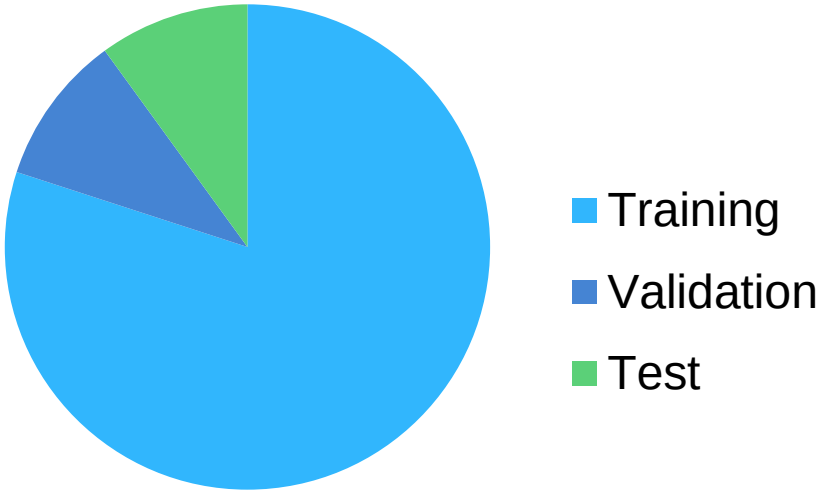
Corpus	Description	Languages	Purpose
Letters	Letters from various authors, different languages	en, de, da/no, others	language identification, author identification
Huffington post	News headlines and abstract from Huffington post, slightly adapted	en	Topic classification
IMBD movie	Movie reviews	en	Sentiment analysis

Manuelle Klassifikation

- Je nach Klassifikationsaufgabe sind die Abweichungen unterschiedlicher menschlicher Klassifizierer unterschiedlich hoch:
  - Sprachenerkennung – sehr gering
  - Themenzuordnung; Genrezuordnung Relevanzeinschätzungen: hoch.

Trainingskorpus - Testkorpus

- Für die Erstellung einer Anwendung muss das Korpus aufgeteilt werden in:
- Testkorpus
- Trainingskorpus
  - Evt. Training/Validation



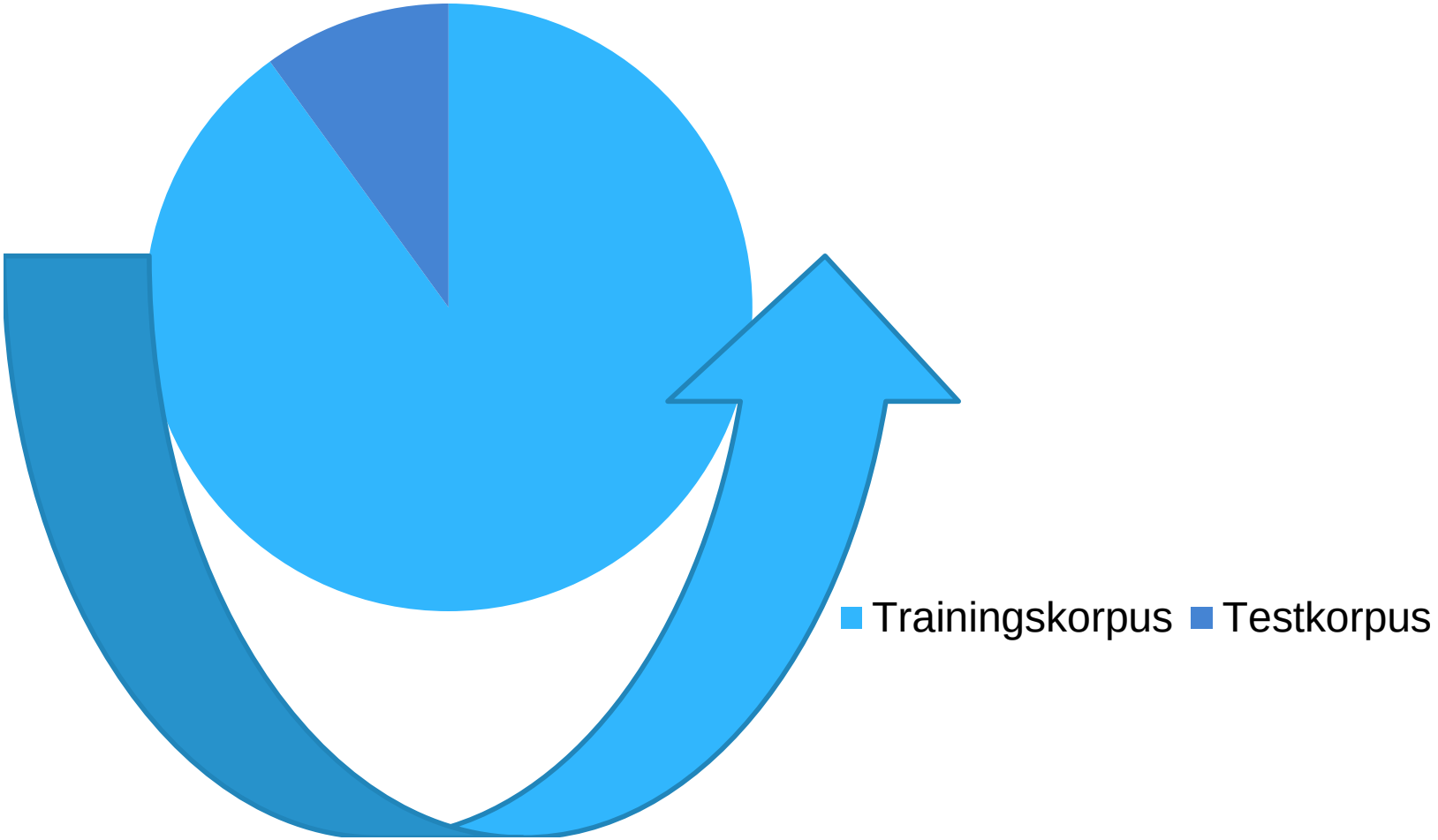
Terminologie

- Training: Zum Trainieren des Modells
- Validation: Optimierung von Hyperparametern oder Iterationskontrolle
- Test: Test auf dem endgültigen Modell

Verhältnis Testkorpus - Trainingskorpus

- i.d.R ist das Trainingskorpus wesentlich größer als das Testkorpus
- Ist das Korpus relativ klein, ist es möglich, Trainingsdaten und Testdaten zu rotieren

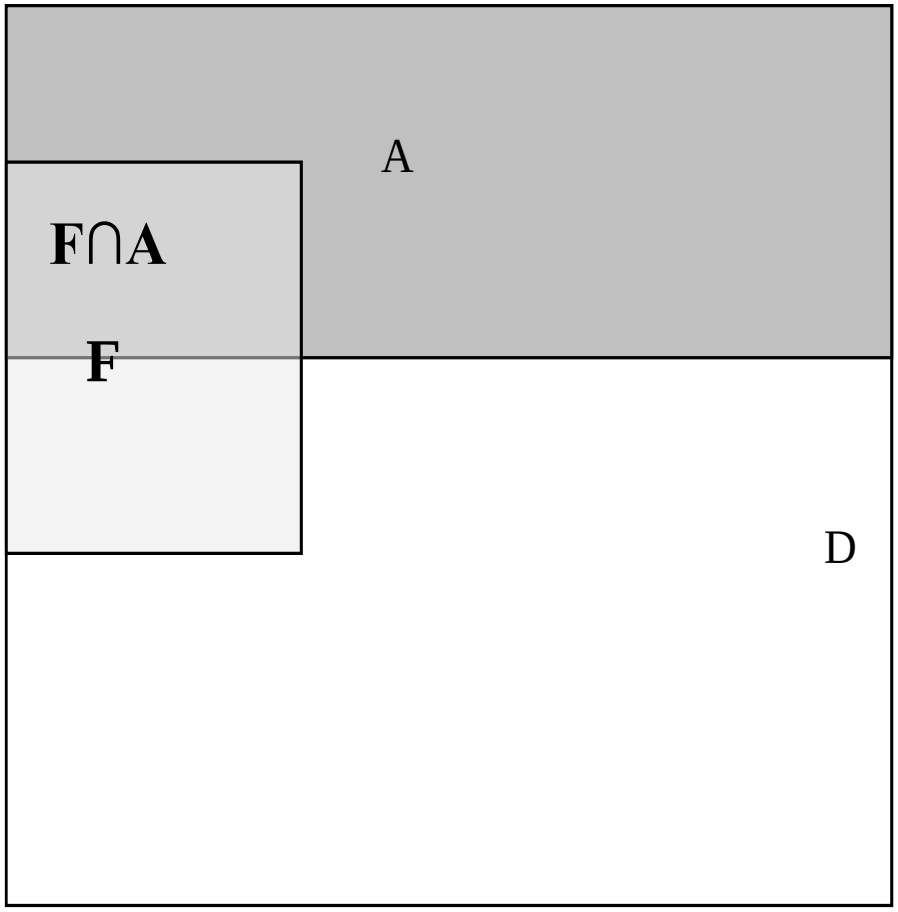
Cross validation



Classification report example

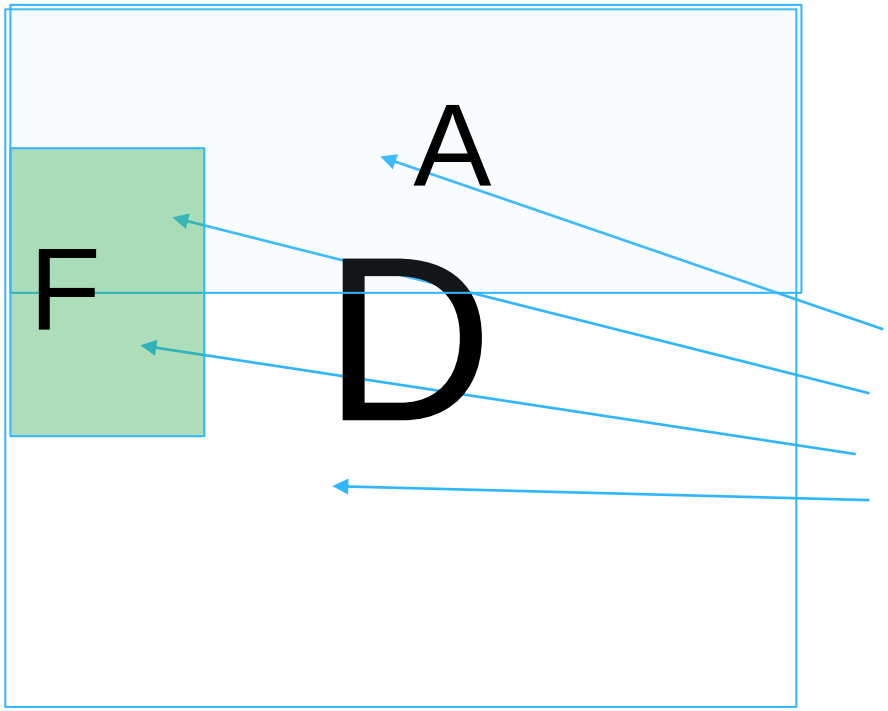
```
#Performance:
Seconds used for training: 55
Seconds used for classification: 12

#Classification report:
              precision    recall  f1-score   support

          da       1.00      1.00      1.00       838
          de       1.00      1.00      1.00      1443
          en       1.00      1.00      1.00      2517
          fr       0.92      1.00      0.96        36
          it       0.94      0.94      0.94        31
     unknown       0.70      0.44      0.54        16

    accuracy                           1.00      4881
   macro avg       0.93      0.89      0.90      4881
weighted avg       1.00      1.00      1.00      4881
```

Trefferquote (Recall) und Genauigkeit (Precision)



- Maß für die Qualität des Retrievals
- D: Alle Dokumente
- A: Relevante Dokument
- F: Gefundene Dokumente
- Recall = F ∩ A/A
- Precision F ∩ A/F

Übertragung von Precision und Recall auf die Evaluierung von Klassifikatoren



- D: Testset
- A: Dokumente, die zur Klasse gehören
- F: Der Klasse zugewiesene Dokumente

- False negatives: ¬ F ∩ A
- True positives: F ∩ A
- False positives F ∩ ¬ A
- True negatives: ¬ F ∩ ¬A

- Recall = F ∩ A/A
- Precision F ∩ A/F

Accuracy

$$\text{Accuracy} = \frac{\text{Richtige Zuweisungen}}{\text{Alle Zuweisungen}}$$

F-measure

Fmeasure: $f = \frac{2\ p*r}{p+r}$

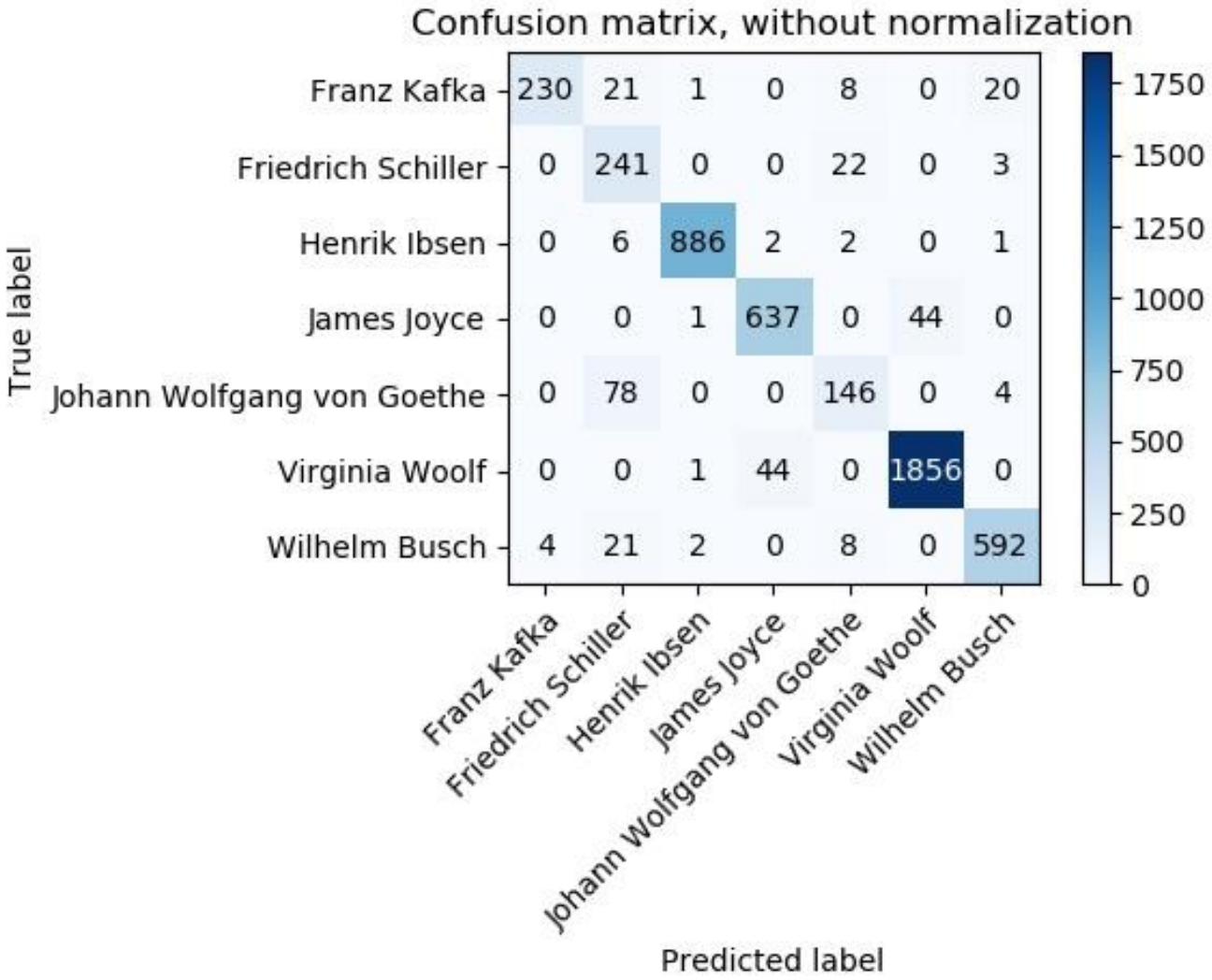
Hier ist p: Precision und r: Recall

(Harmonisches Mittel zwischen Precision und Recall)

Mikro- vs. Makrobewertung

- Mikrobewertung:
  - Gleiches Gewicht für alle klassifizierten Objekte
  - Summiere für alle Klassen $|F \cap A|$, $|F|$ und $|A|$ auf
  - Ermittle Recall und Precision aus den Summen
  - Es gilt: Recall=Precision=Accuracy
- Makrobewertung
  - Ermittle Recall und Precision aus allen Klassen
  - Bilde den Mittelwert

Confusion matrix



Parameteroptimierung

- Grid search and randomized search
- Baysian optimizers

Parameteroptimierung: Grid search

- Search pre-configured parameter settings
- In scikit-learn, pass as dictionary
  - E.g. random forest classifier:

```
parameter_grid = {
            "criterion" : ["gini", "entropy"],
            "max_depth" : [5, 10, 20, 40, 100, 200],
            "min_samples_leaf" : [1, 2, 4, 8]
            }
```

- *Randomized search* is similar, but not alle possible values are used esp. useful for continuous parameter spaces

Evaluation: Clustering

- Interne/intrinsische Kriterien
  - Ähnlichkeit innerhalb des Clusters
  - Unähnlichkeit von Dokumenten in verschiedenen Clustern
  - Bsp. Silhouette-Koeffizient, Davies-Bouldin Index
- Externe Kritierien (z.B. Übereinstimmung mit vordefinierten Klassen)
  - Purity, Normalized mutual information
- Direkte Evaluierung in Bezug auf den Use Case

Scikit-Learn

https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

Purity

$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

Ω: Cluster
ω_k: Elemente in Cluster ω
C: Klassen
c_j: Elemente in Klasse c

Quelle: Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.

Silhouettenkoeffizient

Silhouette des Clusterelements o, welches dem Cluster A zugeordnet ist, Wobei
- B der nächstgelegene Cluster ist
- dist die mittlere Distanz zu den Elemente des Clusters

$$S(o) = \begin{cases} 0 & \text{wenn } o \text{ einziges Element von } A \text{ ist} \\ \frac{\text{dist}(B,o) - \text{dist}(A,o)}{\max\{\text{dist}(A,o), \text{dist}(B,o)\}} & \text{sonst} \end{cases}$$

Silhouettenkoeffizient eines Clusters C:

$$s_C = \frac{1}{n_C} \sum_{o \in C} s(o)$$

Quelle:
wikipedia