Foto: Thomas Josek

Deep Learning for Language Analysis

Deep Learning Introduction – Hands on

Institut für Digital Humanities | Johanna Binnewitt & Valmir Etemi | 09.-10.09.2019

1. Define Research Goal

Research Goal

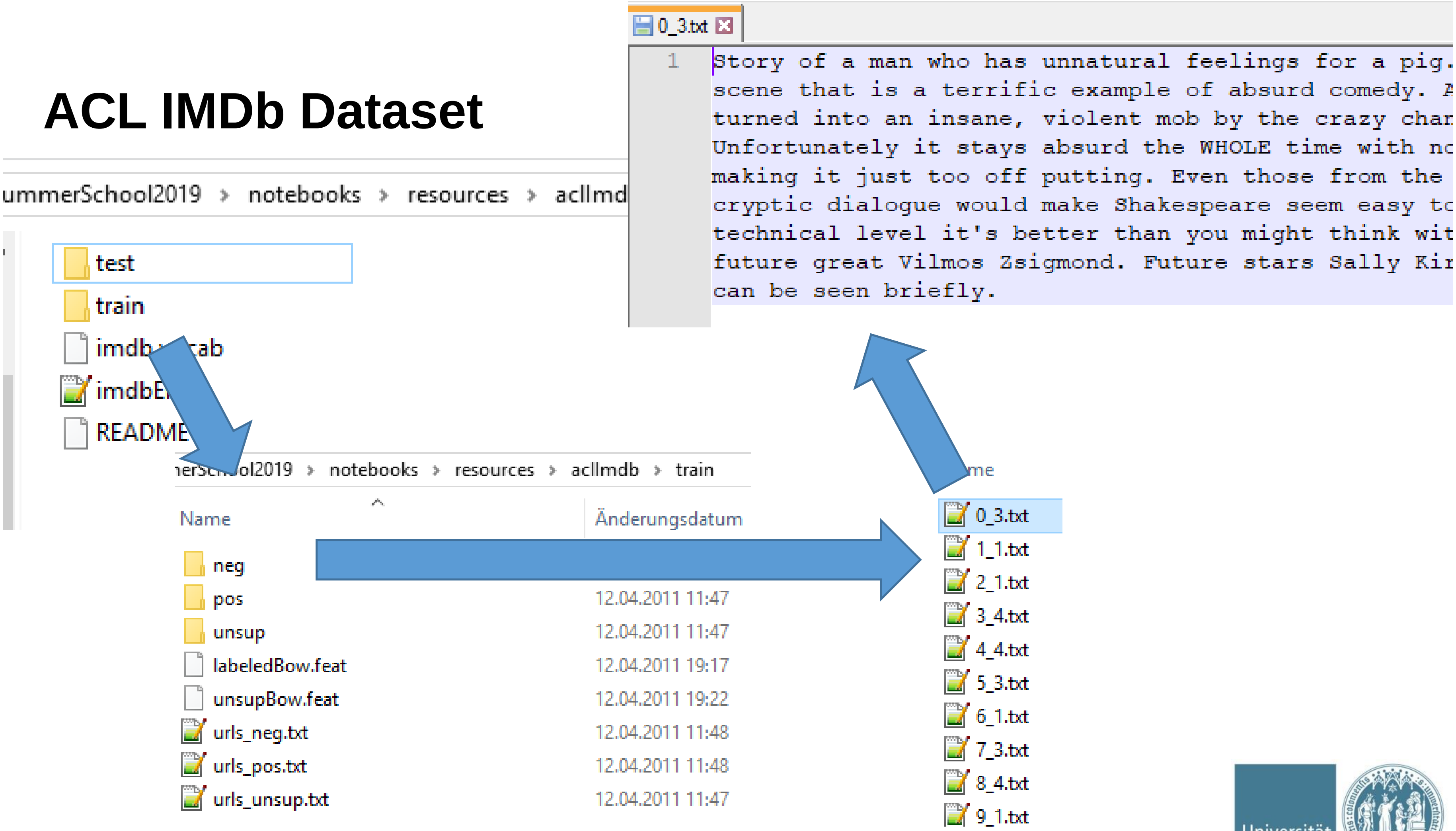
- Binary Sentiment Classification for movie reviews
  - generate a model to classify reviews on movies
  - reviews might be positive or negative (binary classification)
  - use labelled reviews to build model

2. Retrieve Data

- ACL IMDb Dataset
  - Movie reviews divided into positive and negative categories
  - 25.000 entries for training and 25.000 for testing
  - https://ai.stanford.edu/~amaas/data/sentiment/

Text Classification

ACL IMDb Dataset



Name	Änderungsdatum
neg	
pos	12.04.2011 11:47
unsup	12.04.2011 11:47
labeledBow.feat	12.04.2011 19:17
unsupBow.feat	12.04.2011 19:22
urls_neg.txt	12.04.2011 11:48
urls_pos.txt	12.04.2011 11:48
urls_unsup.txt	12.04.2011 11:47

```
Story of a man who has unnatural feelings for a pig.
scene that is a terrific example of absurd comedy. A
turned into an insane, violent mob by the crazy char
Unfortunately it stays absurd the WHOLE time with nc
making it just too off putting. Even those from the
cryptic dialogue would make Shakespeare seem easy tc
technical level it's better than you might think wit
future great Vilmos Zsigmond. Future stars Sally Kir
can be seen briefly.
```

3. Prepare Data

Train/Val/Test Data

- Train Data: to build the model
- Validation Data: to validate/improve the model
- Test Data: to evaluate the final model

- Read .txt-Files from train folder into a Pandas Data Frame
- split Data Frame into train (80%) and validation (20%) set

3. Prepare Data

Prepare Reviews

- Goal: transform reviews (strings) into a suitable model
- Method: Bag of Words
- Keras Tokenizer:
  - Splits each review into tokens
  - Creates a corpus vocabulary
  - num_words: takes only the *n* most frequent tokens into vocabulary for further pre-processing
- Transformed text into a list (but we need vectors)

3. Prepare Data

Binary Vectorization

- Creates a vector for each review
  - Length: num_words
  - Contains: 0 if token is absent in review

    1 if token is present in review

3. Prepare Data

Tf-Idf Vectorization

- Creates a vector for each review
  - Length: num_words
  - Cell entry: Term frequency (w) * Log (document frequency (w))
  - high document frequency → low tf idf score

3. Prepare Data

Prepare Labels

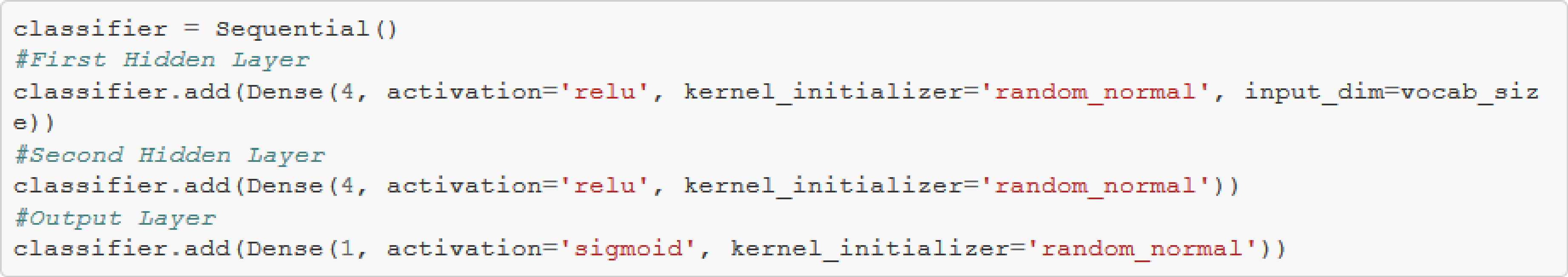
- “neg” and “pos” are categorial labels
- Neural net needs numerical values
- LabelBinarizer
  - Transforms each label into a numerical representation
  - Numerical representation as output of the neural net

4. Explore Data

- How many positive and negative reviews?
- What is the shape of training matrix?
- How does the vocabulary of the Tokenizer looks like?
- What’s the document frequency of a specific token?

5. Model Data

- Goal: Binary Classification
- default neural net uses 3 Dense Layers



```
classifier = Sequential()
#First Hidden Layer
classifier.add(Dense(4, activation='relu', kernel_initializer='random_normal', input_dim=vocab_siz
e))
#Second Hidden Layer
classifier.add(Dense(4, activation='relu', kernel_initializer='random_normal'))
#Output Layer
classifier.add(Dense(1, activation='sigmoid', kernel_initializer='random_normal'))
```

6. Improve Model

- classify some „unseen“ reviews from the validation set
- compare predicted label with true label

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	False Positive

6. Improve Model

- $precision = \frac{true\ positives}{true\ positives + false\ positives}$

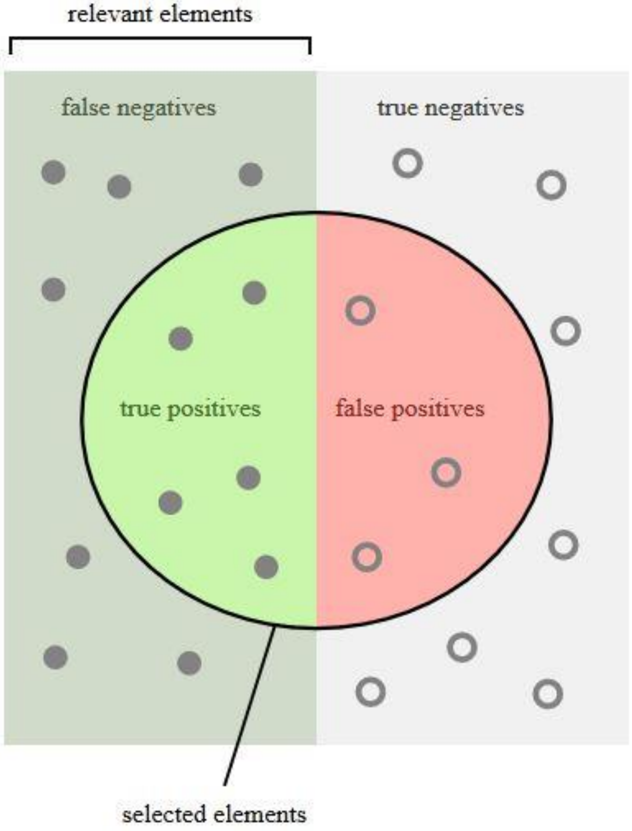
- $recall = \frac{true\ positives}{true\ positives + false\ negatives}$

- $F_1 - Maß = \frac{2 * (precision * recall)}{precision + recall}$

- $F_\alpha - Maß = (1 + \ \alpha^2) * \frac{precision * recall}{(\alpha^2 * precision) + recall}$

- α > 1 → Recall mehr gewichtet

- α < 1 → Precision mehr gewichtet

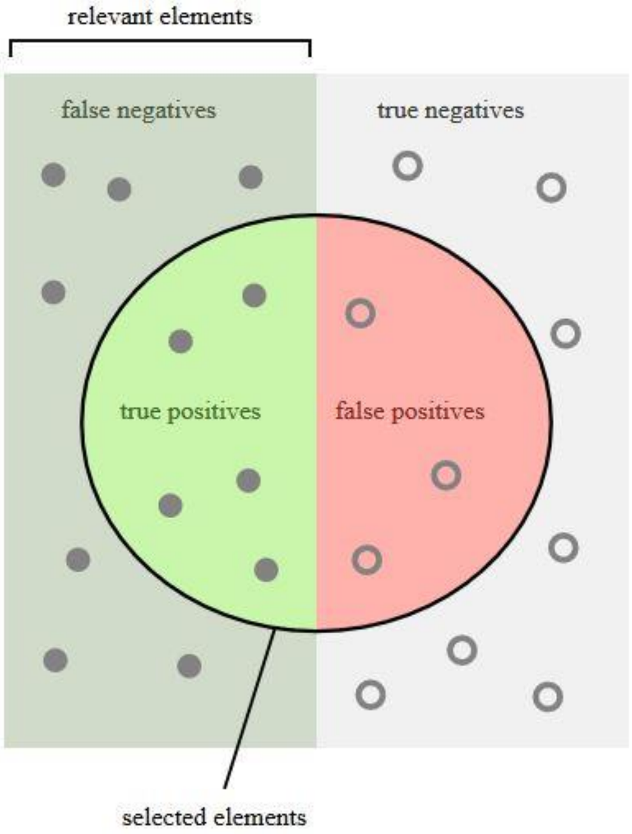


https://en.wikipedia.org/wiki/Precision_and_recall

6. Improve Model

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

- How many of the true-classified objects are actually relevant?

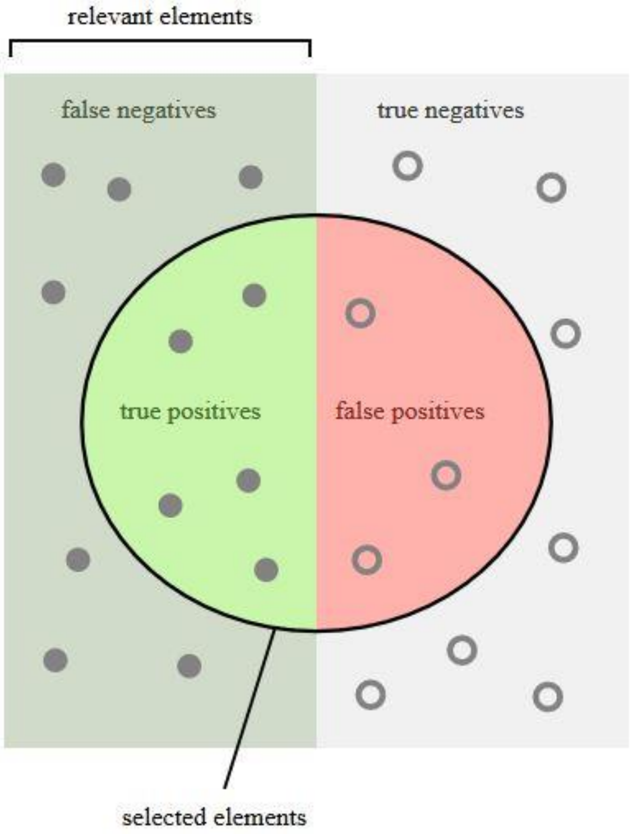


https://en.wikipedia.org/wiki/Precision_and_recall

6. Improve Model

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

- How many of the relevant objects are actually found?



https://en.wikipedia.org/wiki/Precision_and_recall

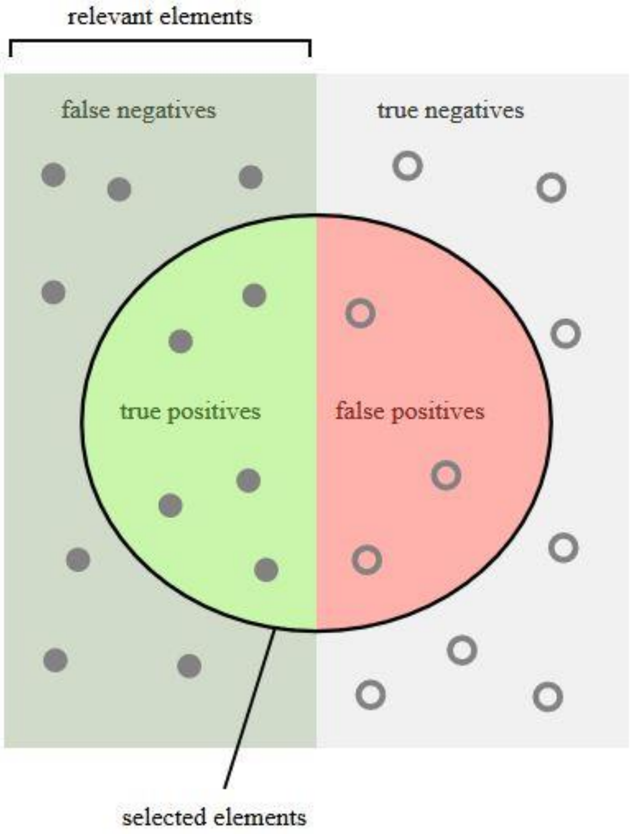
6. Improve Model

$$F_1 - Maß = \frac{2 \ * (precision \ * recall)}{precision + recall}$$

- harmonic mean of precision and recall

$$F_\alpha - Maß = (1 + \ \alpha^2) * \frac{precision \ * recall}{(\alpha^2 * precision) + recall}$$

  - α > 1 → Recall higher weighted
  - α < 1 → Precision higher weighed



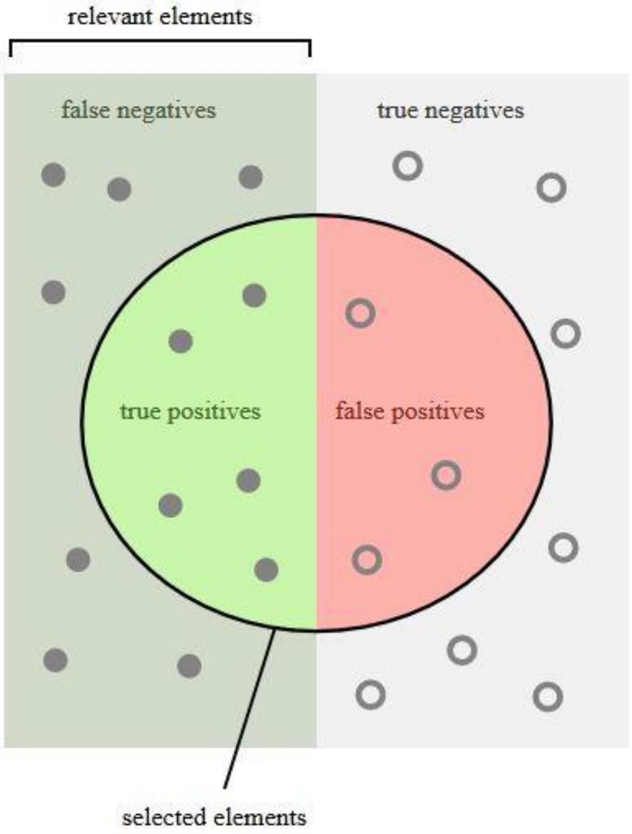
https://en.wikipedia.org/wiki/Precision_and_recall

6. Improve Model

$$accuracy = \frac{true\,positives + true\,negatives}{tp + fp + fn + tn}$$

- How many of my predictions are correct?
- Caution in classifying more than two categories (or imbalanced classes)

→ each **tn** improves the result



https://en.wikipedia.org/wiki/Precision_and_recall

6. Improve Model

- try to improve evaluation results of your neural net
- evaluate your final settings in the **EvaluateSentimentClassification** notebook