Assignment 3: Scientific Area & Similarity Classifier

Advanced Information Retrieval 22/23

Group 23:
Alma Emkic (Developer/Analyser)
Julian Jautz (Architect/AI-Model)
Christina Mandlez (Developer/Visualization)
Paul Scheibelmasser (Project Manager/AI-Dataset)

09.01.2022

https://github.com/pscheibel/AIR_ASSIGNMENT3

Introduction

Research Questions:

Can the affiliation of papers to scientific areas be distinguished automatically by using arxiv category papers as training data?

- Which accuracy can be reached using 5 classes and pytorch.nn?
- How can this application be evaluated meaningfully?

Motivation/Goals:

- Get training data via Arxiv.
    - Automated retrieval of recent papers for various categories
- Implement Network classifying scientific belonging
    - Extract nouns via text processing (textblob)
    - Creation and training of Classification Network based on nouns
- Classify and Compare any Papers (URLs).
    - Input:
        - Papers the Network will be applied on
    - Output:
        - Category per Paper
        - Plot of training/network statistic
        - Plots of Similarity

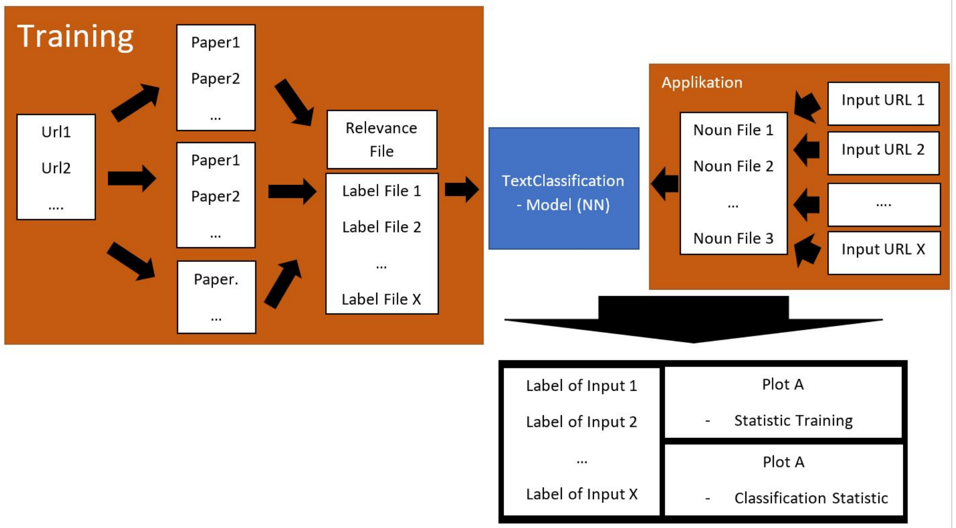
Data

- papers from https://arxiv.org/
  - Retrieves automatically x newest papers of configured categories.
  - Downloads content of URL and processes its nouns into labelled files, which are stored via caching.
  - Creates labelled dataset for machine learning (Categorization)

  - cs
    - https://arxiv.org/list/cs/pastweek?show=1000
  - q-bio
    - https://arxiv.org/list/q-bio/pastweek?show=1000
  - physics
    - https://arxiv.org/list/physics/pastweek?show=1000
  - eess
    - https://arxiv.org/list/eess/pastweek?show=1000
  - econ
    - https://arxiv.org/list/econ/pastweek?show=1000

Methods

- Technologies
    - Python (Pycharm)
    - Neural Network
- Algorithms
    - Noun extraction
    - Neural Network
        - Loss Function: CrossEntropyLoss
        - Optimizer: SGD
- Libraries
    - Textblob
    - PyPdf2
    - torch
    - urllib.request
    - BeautifulSoup



Training

Url1
Url2
....

Paper1
Paper2
...

Paper1
Paper2
...

Paper.
...

Relevance File

Label File 1
Label File 2
...
Label File X

TextClassification - Model (NN)

Applikation

Noun File 1
Noun File 2
...
Noun File 3

Input URL 1

Input URL 2

....

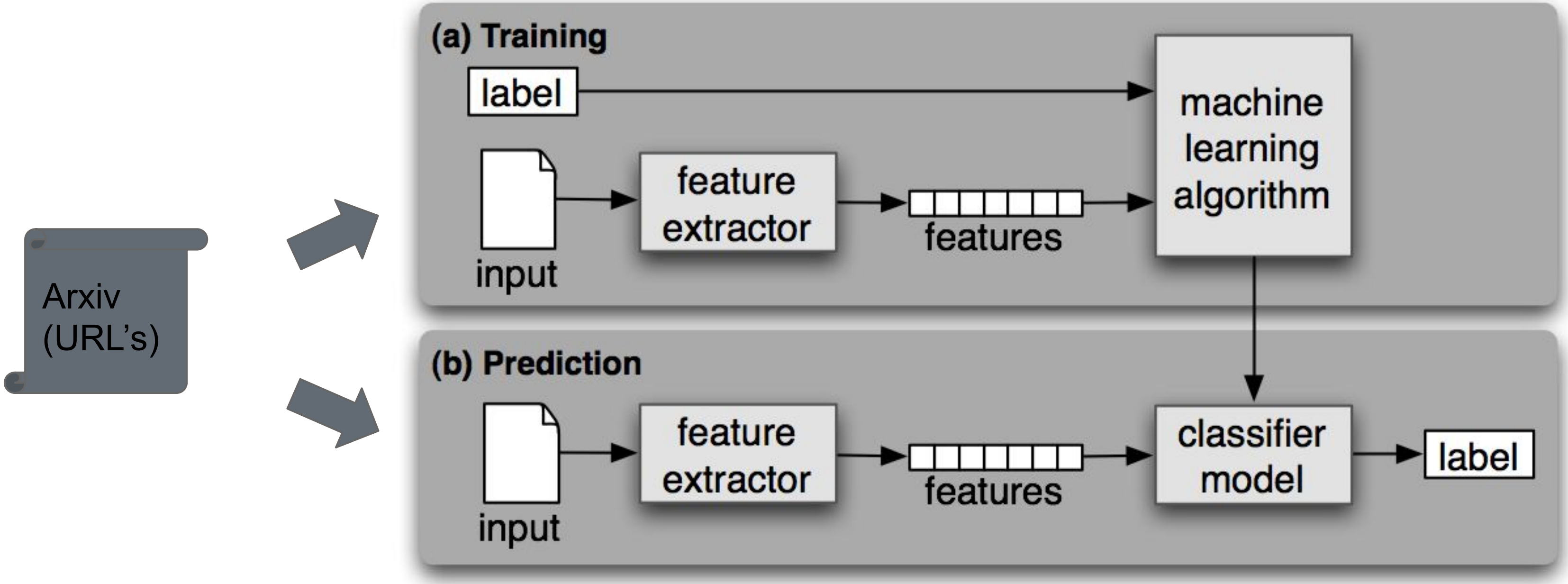
Input URL X

Label of Input 1
Label of Input 2
...
Label of Input X

Plot A
- Statistic Training

Plot A
- Classification Statistic

Methods



Arxiv (URL's)

https://www.nltk.org/book/ch06.html

Results – Analysis and Interpretation

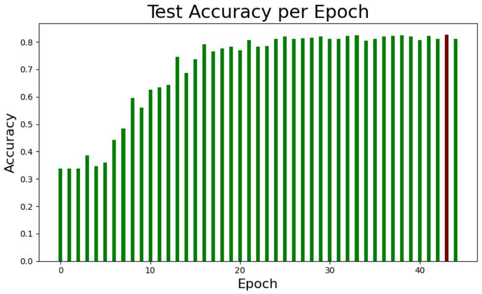
Literature comparison:

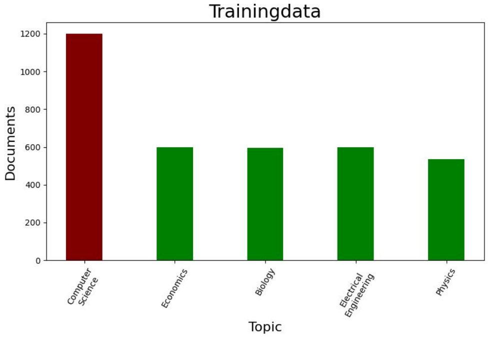
- How I achieved 90% accuracy on a text classification problem with ZERO preprocessing
  - Accuracy: 90%
  - BERT sentence embeddings
  - used Spark NLP
  - 4 categories
- Text Classification with TF-IDF, LSTM, BERT: a comparison of performance
  - TF-IDF (97.9%)
  - Recurrent Neural Networks (94.6%)
  - Bert Language Model (96.6%)
  - 5 categories

Our Application:

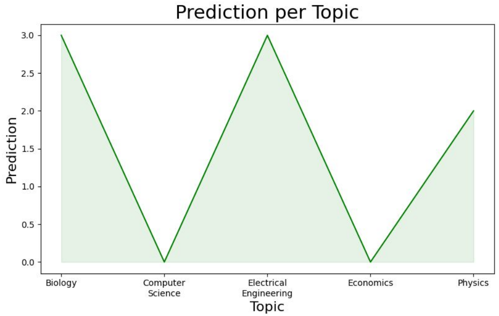
- Accuracy > 80%
- used pytorch.nn
- Optimizer SGD
- CrossEntropyLoss
- 5 categories

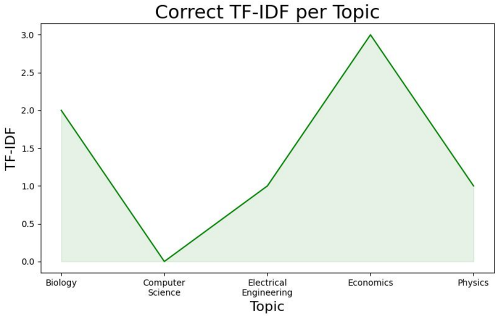
Results – Analysis and Interpretation





- Comparison
  - TF_IDF
    - term frequency per scientific category
    - classifies ~53% of papers correctly
  - Pytorch.nn
    - classifies ~57,33% of papers correctly





Conclusion

Bias:

- too much data in one field biases model
  - training leans on Computer Science (more CS papers than other fields)
- common paper expressions, that occur in all areas are not filtered

Limitations:

- not able to classify mathematics
- not enough data
  - current status is 1200 data sets
  - more data would be better (50000 data sets)