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)

https://github.com/pscheibel/AIR_ASSIGNMENT3

Introduction

Research Questions:

Can affiliation of papers to scientific areas be distinguish 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 meaningful evaluated?
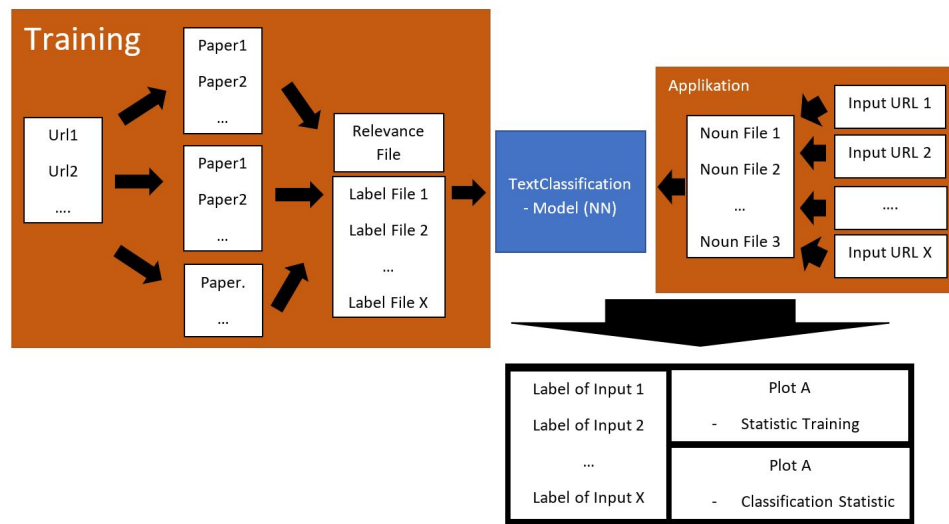
Motivation/Goals:

- 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
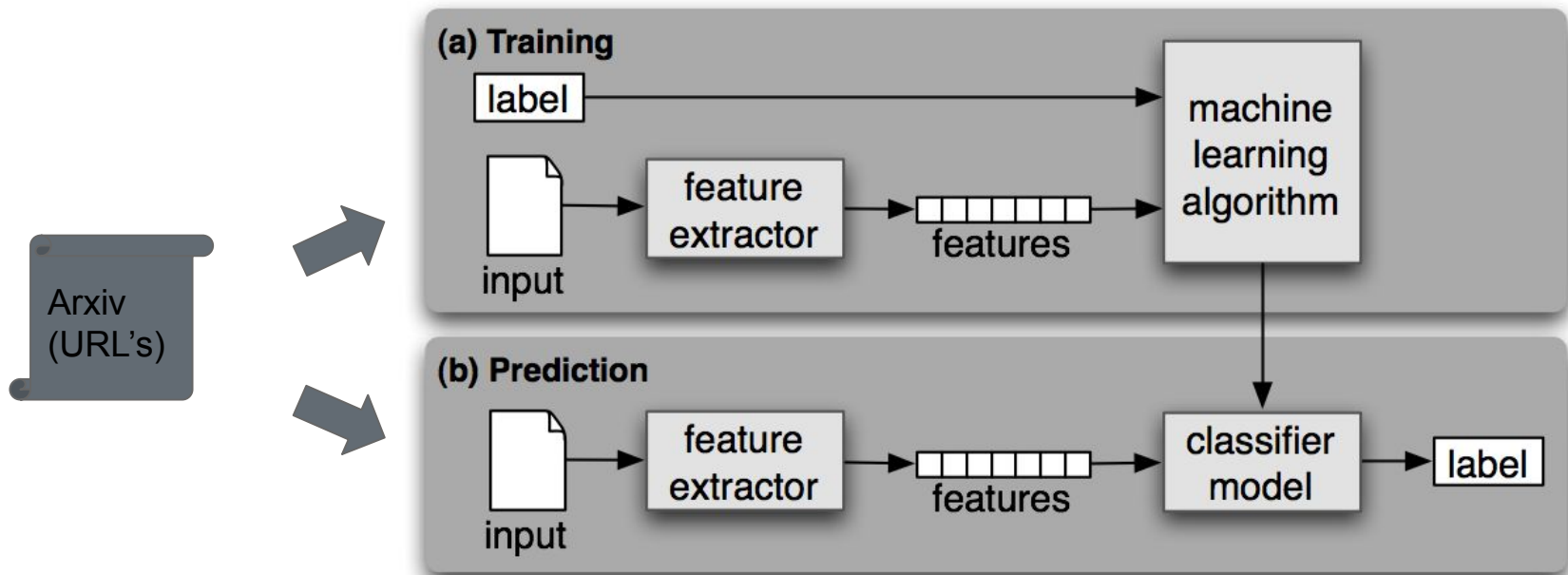   - Plots of Similarity

Data

- papers from https://arxiv.org/
  - Retrieves automatically x newest papers of configured categories y 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

Methods

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

Results – Analysis and Interpretation
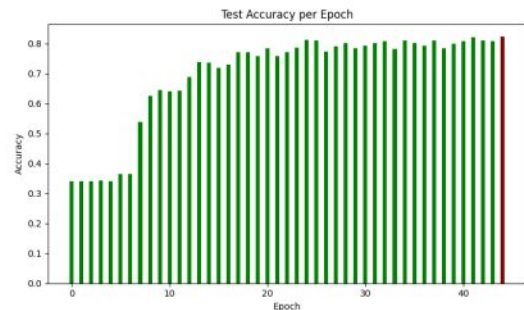
Literature comparison:

* **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
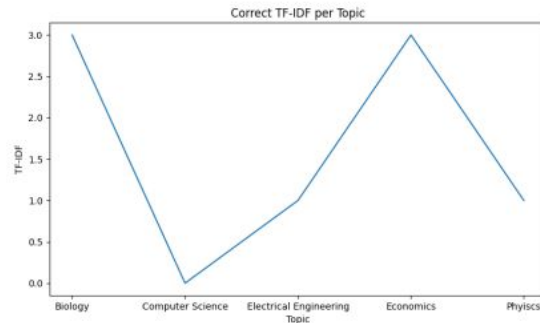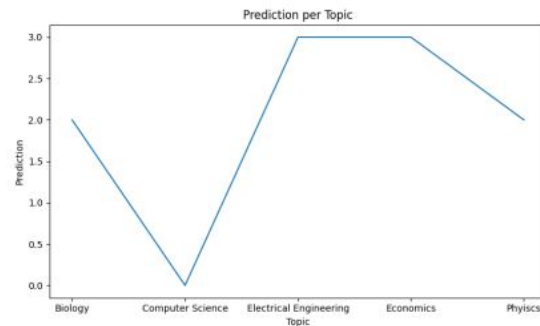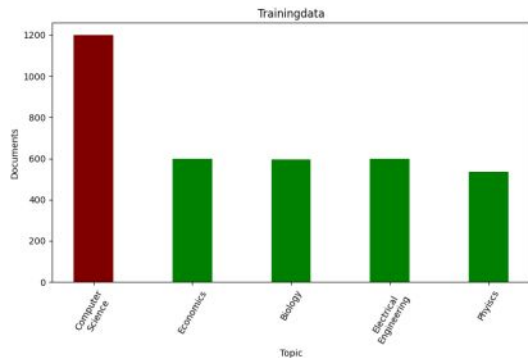* 5 categories

Results – Analysis and Interpretation

Test Accuracy per Epoch

Trainingdata

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

Prediction per Topic

Correct TF-IDF per Topic

Conclusion

Bias:

- ❖ Computer Science - training leans on Computer Science
- ❖ common paper expressions that occur in all areas are not filtered

Limitations:

- ❖ we have not managed to classify mathematics
- ❖ not enough data, current status is 1200 data, but 50000 data would be better
- ❖ time