Deep Learning in NLP

Project term paper

Contents

1 Changelog

- **03.03.2021**: Added information on the **structure** of the paper in Sec. 2.

2 General guidelines

- **Topic**: You can pick a topic of your choice (see Sec. 3 for examples)
- **Groups**: You can do the project in groups (max. 3 members).
- **Format**: Please use the ACL 2020 stylesheet (Latex, Word).
- **Length**: 4-5 pages for undergrad students, 7-10 pages for master students.
- **Code**: Documented code[1] and running/installation instructions make part of the deliverables.
- **Evaluation**:
  - Clarity of writing / presentation
  - Soundness / correctness
  - Empirical evaluation

[1]The code should be documented the standard way: you should provide docstrings/comments which explain what it does and why, especially for more complicated chunks of code.

  - Dataset / effort
  - Code quality / documentation

- **Structure**:
  - The term paper should provide a description of the selected task, the implemented model, the dataset, the performed experiments and the corresponding results, etc.
  - The point of the paper is **not** to explain the implementation. It's the other way around – the code should provide an implementation of the model (in particular) described in the paper. You can add a separate section with high-level description of the implementation, if needed, but otherwise avoid refering to the types/functions/classes/methods defined in the code.
  - Look at research articles on a similar topic to better understand how a term paper should look like.

3 Possible project topics

Examples of past project topics:

- Character recognition for Kyrillic characters
- Text generation based on Dickens' novels
- Sentiment analysis on mined Twitter data
- Implementation of Flappy Birds using genetic algorithms
- Topic classification of speeches from the German parliament
- Detection of English puns (SemEval-2017 Task 7)

Other possible topics, based on the code developed during the practical sessions:

- Handling the BERT tokenization mismatches in the dependency parser, hyperparameter fine-tuning, and comparison with state-of-the-art
- Extending the dependency parser with the prediction of morphological features (the `FEATS` column, see the CoNLL-U format)
- Improving the toy RNN encoder/decoder translation system with, for example: attention mechanism (see the appendix in the specs), batching (to speed up training), beam search decoding, proper evaluation method, or other techniques that can be found in the literature

4 Frequently asked questions

- **Which language should the report be written in?**
  Both English and German are fine.