Learning sentence representations from natural language inference data

SMNLS: Practical I

March 2019

1 Introduction

The first practical of the Statistical Methods for Natural Language Semantics course concerns learning general-purpose sentence representations in the natural language inference (NLI) task. The goal of this practical is threefold: to *implement* three neural models to classify sentence pairs based on their relation, to *train* these models using the Stanford Natural Language Inference (SNLI) corpus [1] and *evaluate* the trained models using the SentEval framework [2].

NLI is the task of classifying entailment or contradiction relationships between premises and hypotheses, such as the following:

1. *Premise* Bob is in his room, but because of the thunder and lightning outside, he cannot sleep.
2. *Hypothesis 1* Bob is awake.
3. *Hypothesis 2* It is sunny outside.

While the first hypothesis follows from the premise, indicated by the alignment of 'cannot sleep' and 'awake', the second hypothesis contradicts the premise, as can be seen from the alignment of 'sunny' and 'thunder and lightning' and recognizing their incompatibility.

2 SNLI corpus

The SNLI corpus (version 1.0) is a collection of 570k human-written English sentence pairs manually labeled for balanced classification with the labels entailment, contradiction, and neutral, supporting the task of natural language inference (NLI), also known as recognizing textual entailment (RTE). Here you can find all the information you need about this corpus:

`https://nlp.stanford.edu/projects/snli/`

3 Models

Your task in this assignment is to replicate some of the results reported by Conneau et al. [3]. An overview of the architecture is shown in Figure 1.

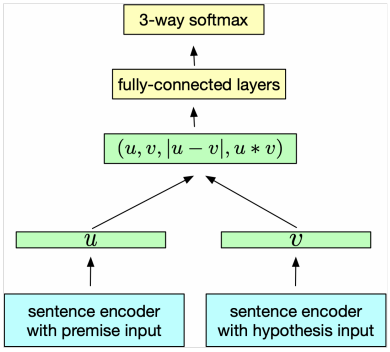

Figure 1: General architecture for sentence classification

For the sentence encoder blocks, you need to implement these four different models:

1. Baseline: averaging word embeddings to obtain sentence representations. You should use the 300-dimensional GloVe embeddings trained on Common Crawl 840B: `https://nlp.stanford.edu/projects/glove/`;
2. Unidirectional LSTM, where the last hidden state is considered as sentence representation (see Section 3.2.1);
3. Simple bidirectional (bi-) LSTM, where the last hidden state of forward and backward layers are concatenated as the sentence representations;
4. Bidirectional LSTM with max pooling applied to the concatenation of final hidden states from both directions to retrieve sentence representations (see Section 3.2.2).

To achieve comparable results and avoid extensive hyperparameter search, we advise you to use the hyperparameter settings as described in Section 3.3.

4 Evaluation

The SNLI dataset consists of three splits, (1) train: which you should use to train your models, (2) dev: which you should use can use for hyperparameter tuning, and (3) test: which you should use to evaluate the models. You should evaluate your models on the SNLI task using macro and micro accuracy metrics (see section 5 of the paper).

In addition, you need to use SentEval to evaluate the sentence representations obtained from each of the models. SentEval, the Facebook evaluation toolkit for sentence embeddings, is a library for evaluating the quality of sentence embeddings by applying them on a broad and diverse set of downstream tasks called "transfer" tasks. The reason they are called transfer tasks is that the sentence embeddings are not explicitly optimized on them.

The link to the SentEval repository on github:

- https://github.com/facebookresearch/SentEval

A tutorial on how to install and use SentEval to evaluate GloVe, when GloVe word vectors are averaged for the words in a sentence to compute the sentence representation is available here:

- https://uva-slpl.github.io/ull/resources/practicals/practical3/senteval_example.ipynb

5 Practical Matters

1. You can either use tensorflow or pytorch for this assignment. You can and are encouraged to use high level APIs for efficiency reasons.
   Some usefull resources:
   - `https://www.tensorflow.org/tutorials`
   - `https://pytorch.org/tutorials/`
   - `https://paperswithcode.com/`
   - `http://ruder.io/text-classification-tensorflow-estimators/`
2. Use tensorboard to visualize and monitor the training process of your models and include the plots from tensor board in your report.
   - `https://www.tensorflow.org/guide/summaries_and_tensorboard`
   - `https://github.com/lanpa/tensorboardX`
3. To download and preprocess the data, you can use the code available here: `https://github.com/brmson/dataset-sts/tree/master/data/rte/snli`
4. If the original link for downloading the dataset doesn't work use one of these links:
   - `https://www.kaggle.com/stanfordu/stanford-natural-language-inference-corpus/version/1`
   - `http://www.cl.cam.ac.uk/~jm864/snli_1.0.zip`
5. Prepare a tiny version of the dataset for debugging your implementations, and make sure your implementations work fine before starting the training process on the original dataset. Make sure you can over-fit your models on the tiny dataset (As a debugging technique).

6. Have separate modules for training and inference phases. You should have an interface similar to this:
   For training:

   ```
   python train.py <model_type> <model_name> <checkpoint_path> <train_data_path>
   ```

   For evaluation:

   ```
   python eval.py <checkpoint_path> <eval_data_path>
   ```

   For inference:

   ```
   python infer.py <checkpoint_path> <input_file> <output_file>
   ```

7. Prepare a Jupyter notebook, where you can load a trained model and demo how it works for different examples.
8. Do some error analysis.

5.1 Reading resources

Please, read these papers thoroughly before starting to work on the practical:

- A large annotated corpus for learning natural language inference [1]
- Supervised learning of universal sentence representations from natural language inference data [3]

You are very much encouraged to have group discussions about these papers to fully understand the task and the models.

In addition, if you are not familiar with neural network models for NLP you can take a look at these blog posts:

- `http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/`
- `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`
- `http://ruder.io/deep-learning-nlp-best-practices/`

5.2 Report

Prepare a one page report on the results you get from each of the models you implement. You shouldn't write any explanations in this report. Just try to find cool ways of visualizing your results!

5.3 Submission

You should do this practical individually and submit your report in a pdf on Canvas with the title **SMNLS-Practical1-FullName**. Submission deadline for the report is **23:59 on Friday, 19 April**. In addition, upload your codes on a github repository and put a link to the repository in your report. Also, add the instructions on how to run your codes for each part of the practical to the github ReadMe.

5.4 Grading

The delivery of the project will be in person where you demo your project. You should show the general performance of the models in terms of numbers and plots, and also show the output of your models for different examples. We will go through your results and implementations together and ask a couple of questions about your models and the results. You should be able to explain in detail how each of the models work both from the theoretical and implementation point of view. Also, you need to be able to compare the models in terms of their performance on this task and make sound arguments about why you think one is getting better or worse results than the other ones. You will be asked to analyse the results and share your interesting insights with us.

References

[1] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[2] Alexis Conneau and Douwe Kiela. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

[3] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.