Assignment 3

For this assignment, you are going to train and test two BERT models for SRL: (1) a baseline model and (2) a more advanced model. You will submit your work in the form of a guided notebook explaining the steps you took and choices you made. Please store your trained models, so you can reuse them for the take-home exam you will complete at the end of the course.

BERT models

You are going to fine-tune BERT to detect and label arguments (in a single step). For this assignment, you do not have to take care of detecting predicates. You can use the predicates in the gold data. As with the previous assignment, please use the Universal Proposition Bank for English (v1.0) as your training and test data. To understand how you can use BERT for sequence tagging, please feel free to use this tutorial:
https://github.com/huggingface/notebooks/blob/main/examples/token_classification.ipynb

The tutorial shows you how to fine-tune BERT on simple sequence tagging problems (e.g. NER). SRL is more complicated, as it is a structure prediction problem (rather than a pure sequence tagging problem). The main challenge of this assignment is to find a solution for representing the relation between the sentence and the predicate under consideration.

The tutorial works with tasks that use BIO token labels to deal with span labeling problems. While argument identification and labeling is technically a span labeling problem, we do not expect you to convert the data from argument head labels to argument spans. Please simply work with the token labels in the data.

Baseline model

Whenever we create a new model, it is good practice to compare it to a baseline model. The goal is to predict the correct argument labels for each token in a sentence given a particular predicate in the sentence (attention: one sentence can have multiple predicates). In this assignment, we are particularly interested in how we represent the model input in such a way that we can label the tokens in a sentence with respect to a particular predicate. As a baseline model, you are going to use a simple approach inspired by Shi and Lin (2019):

[CLS] Barack Obama went to Paris [SEP] went [SEP]

This approach only requires a minimal modification of the code provided in the tutorial; you will have to make sure that the output you receive from the model can be matched against your gold data so you can evaluate the predictions. Please note that the approach described by Shi and Lin (2019) is slightly more complicated than what we propose here; they add a predicate indicator variable (indicating whether the current token is the predicate or not) to the encoder output. We do not expect you to implement this. Simply provide the input as outlined above.

Advanced model

The baseline approach has several limitations (consider sentences in which the same predicate token appears more than once). As a more advanced model, you are going to implement a version of your baseline that represents the relation between predicate and sentence in a better way. You can take inspiration from the approaches introduced in the lectures (e.g. Zhou and Xu 2015 for SRL) or from approaches to negation detection and scope resolution (Khandelwal, et al. 2020) or come up with your own approach to marking the predicate in a more precise manner.

Note: We do not expect you to add features to the input representation. You can limit your experiments to modifying the input text the model receives.

Evaluation

Your notebook has to contain an evaluation of both models in terms of precision, recall, and f1 score for argument detection and labeling. Please provide (1) an aggregated score and (2) scores per argument label for your baseline model and your advanced model.

Notebook format and requirements

Components

Your notebook should contain all the steps taken to carry out your experiment:

(1) Extracting the data from the CoNLL format and converting it to a BERT input format (one instance is a sentence-predicate combination)
(2) Creating input for the BERT model (baseline + advanced model)
(3) Finetuning your models (using the training split)
(4) Evaluation of your models on the test split

Code format

Please do not define functions in your notebook. Rather, place them in utils scripts and import them on top of your notebook. Please avoid using global variables in your functions (attention: the tutorial contains global variables). Your notebook should be easy to read.

Descriptions and explanations

Your notebook has to contain the following information

- A clear description of the steps you are carrying out
- A description of the limitations of your baseline model
- A description of how your more advanced model addresses the limitations of your baseline model
- A description of the limitations of your advanced model
- A results table (please save the notebook output rather than have us rerun every step)
- A brief description of your results (how does your advanced model compare to the baseline?) (A couple of sentences are enough.)
- Conclusions and directions for future work

Submission

Please submit a link to your code repository (make sure we can access it). If you are very uncomfortable with putting your code online, you can submit a .zip folder with your code. Please make sure it does not contain data or models. Your assignment will be graded with 0 if your .zip folder contains data or models. Note: you are not allowed to modify your code after the assignment deadline (we will see edit dates in code submitted online).

Grading

For a passing grade, the following components have to be present:

- Implementation (50%)
    - Baseline implementation
    - Advanced implementation
    - Evaluation
- Descriptions and explanations (50%)
    - Description of both models and their limitations (in particular with respect to the input representation)
    - Description of the results
    - Conclusions and directions for future research

References

Khandelwal, A. and Sawant, S., 2020, May. NegBERT: A Transfer Learning Approach for Negation Detection and Scope Resolution. In Proceedings of the Twelfth Language Resources and Evaluation Conference (pp. 5739-5748).
http://www.lrec-conf.org/proceedings/lrec2020/pdf/2020.lrec-1.704.pdf

Shi, P. and Lin, J., 2019. Simple bert models for relation extraction and semantic role labeling. arXiv preprint arXiv:1904.05255.
https://arxiv.org/pdf/1904.05255.pdf

Zhou, J. and Xu, W., 2015, July. End-to-end learning of semantic role labeling using recurrent neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 1127-1137).
https://aclanthology.org/P15-1109.pdf