Assignment 1
NLP 203: Natural Language Processing III

University of California Santa Cruz

All assignments are to be completed individually. You may discuss with the TAs and the instructor, but you may not receive help from anyone else.

Instructions. Submit a zip or tgz file containing your writeup, and output and code to Canvas. Although not required, we encourage you to use LaTeXto typeset your writeup.

Neural Machine Translation

The goal of this assignment is to experiment with automatic machine translation using various deep learning models. For this assignment, we will use the IWSLT'13 dataset for French-English **fr-en**. The dataset can be downloaded from here[1]. The IWSLT datasets were created from TED talks using transcripts and their manual translations. Please refer to the README included in the dataset directory for information about each file.

We will use the Fairseq library for all experiments. This library comes with documentation and examples of machine translation [2] for various datasets and neural architectures.

split	# sentence pairs
train	153,300
dev	887
test	1,664

Table 1: Statistics for fr-en IWSLT'13

Refer to the data preparation scripts to clean and tokenize the data, and create train, dev and test splits (including BPE preprocessing). For this assignment, we will use the dev2010 and test2010 files to create the dev and test splits, respectively. This should give you the data statistics in table 1. Please use the dev data for hyper-parameter tuning, and report results on the test split. For all experiments, use SacreBLEU [1] to report the performance of each of your models.

[1] https://drive.google.com/drive/folders/1WGxy_Cm6CEEl74jeNrydZ5qyCYYzsE3f?usp=share_link

[2] https://github.com/pytorch/fairseq/tree/main/examples/translation#training-a-new-model

Part 1 [15 points]

Train a CNN and a transformer encoder decoder model using Fairseq for the fr-en language pair. As in the original Transformer paper, [4], use byte-pair encoding to encode the sentences. Generate translations for the sentences in the test splits.

Questions

1. Report BLEU scores (untokenized, using SacreBLEU).
   Expect your BLEU scores to be 30 or higher.
   (Take note of the number of tokens after BPE tokenization in each split to compare to your models in part 2, when BPE is not used.)

Part 2 [15 points]

Next, turn off the BPE tokenizer, and retrain your Transformer model from Part 1 on whole words tokenized using the Moses tokenizer. Generate translations for the sentences in the test split. (Remember to save these translations to look at them later.)

Questions

1. What are the number of tokens in each split without running BPE? How does this compare with in Part 1 using BPE?
2. Report the BLEU score (untokenized, using SacreBLEU).
3. Find and report at least 3 example sentences from the development set which have better or worse translations than in the model from Part 1. (You could use Google Translate to get a sense of the translation quality, if you are unfamiliar with the language.)

Part 3 [15 points]

In this part, the goal is to improve your model from Part 1. You may try one or more of the following, or any other ways you can think of to improve the BLEU scores [3]:

- use shared source and target embeddings in your transformer model. (Refer to [2])
- try out different number of layers, or tune dropout.
- tune number of operations in BPE, or, the dropout in BPE, or, learn BPE on the concatenation of the training text. (Refer to subword-nmt[3])

[3] `https://github.com/rsennrich/subword-nmt`

Questions

1. Describe in detail everything you tried, and the performance in each case, in terms of BLEU scores.
2. What is the best performance you could get using the techniques you tried. Report your best BLEU score (untokenized, using SacreBLEU).
3. Find and report at least 3 examples from the development set which have better translations than in the model from Part 1. Find one example of a worse translation than the model in Part 1.

Suggestions To cut down on training time, train each of your models for no more than 10 epochs. This should take about 30-40 minutes on the GPU server. Please be sure to start early, to be able to run all your experiments.

Deliverables In the writeup, be sure to fully describe your models, experimental procedure and hyperparameters used, along with answers to the questions in each part. Submit all your code, and the translation outputs from your best model in the zipfile containing your writeup.

References

[1] Matt Post. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*, 2018.

[2] Ofir Press and Lior Wolf. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*, 2016.

[3] Rico Sennrich and Biao Zhang. Revisiting low-resource neural machine translation: A case study. *arXiv preprint arXiv:1905.11901*, 2019.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.