Assignment 2

Due: October 12th, 2020

1 Introduction

In this assignment, you will explore strategies to build competitive neural machine translation (NMT) systems for low-resource languages. NMT models are the state-of-the-art method for translation between many common languages, backing popular translation engines such as Google Translate. However, it lags behind the traditional statistical machine translation models when the available parallel data is limited.

2 Data set

We will use the multilingual translation data set from [8]. It contains parallel data extracted from TED talks from 58 languages to English. The raw data can be downloaded from the link in the homework github repo. We will focus on translation to and from English for two low-resource languages: `aze` (Azerbaijani) and `bel` (Belurasian).

3 Baselines

In this assignment, you can build the NMT system on top of an open-source toolkit fairseq. We provide some example preprocessing and training scripts for fairseq, but you can use any toolkit you like. Another more light-weight NMT repository you could use is JoeyNMT.

The first step for the assignment is to train an NMT system using only parallel data from the low-resource languages. We provide a complete data preprocessing script and training script for running the baseline model for one of the low-resource languages `aze` at the homework github repository.

You should read the provided scripts to understand the data processing pipeline, the training and model evaluation method. With slight modification to the provided scripts, you should be able to run the baseline for both aze and bel. Your experiment should match the results in Table. 1. Here we experiment with translation both from Azerbaijani to English (aze-eng) and English to Azerbaijani (eng-aze).

4 Multilingual training

Note that since the languages we consider have very limited amount of parallel training data, the NMT model performs really bad with BLEU scores less than 10. This is a known issue for neural models in general - they are much more data hungry than the statistical methods. Luckily, we can use multilingual training to boost the performance of these low resource languages. In this section, you will learn to use data from the high resource related languages to improve the NMT performance.

Language	BLEU
aze-eng	2.7
eng-aze	1.7
bel-eng	1.94
eng-bel	1.09

Table 1: Baseline bilingual results

Language	BLEU
aze-eng	12.34
eng-aze	5.45
bel-eng	17.55
eng-bel	9.05

Table 2: Multilingual results

A closely related language for aze is tur (Turkish), which has much more parallel data (about 200k sentences in the TED corpus). Essentially, we just need to train a standard NMT model using concatenated data from both aze and tur. The idea is that the knowledge about tur, a closely related language with a lot of similar vocabularies, can help the model to translate aze.

We provide training and evaluation scripts for doing multilingual training for aze using fairseq. For bel, you could use data from rus (Russian) as the related high-resource language. You would be able to match the results in Table. 2.

5 Improving Multilingual Transfer

5.1 Data Augmentation

Extra monolingual data is often helpful for improving NMT performance. Specifically, back-translation [2, 9] has been widely used to boost the performance of low-resource NMT. A closely related method, self-training [3], is recently proven to be effective as well. Recently, several methods have been proposed to combine different methods of using monolingual data with multilingual training. [15] explores several different strategies for modifying the related language data to improve multilingual training. [10] adds a masked language model objective for monolingual data while training a multilingual model.

5.2 Choosing Transfer Languages

For the provided multilingual training method, we simply use a closely related high-resource language as a transfer language. However, it is likely that data from other languages would be helpful as well. [5] has done a systematic study of choosing transfer languages for NLP tasks. There are also methods designed to choose multilingual data for specific NLP tasks [12, 11].

5.3 Better Word Representation or Segmentation

Vocabulary differences between the low-resource language and its related high-resource language is an important bottleneck for multilingual transfer. [13] propose a character-aware embedding method for multilingual training. For morphological rich languages, such as Turkish and Azerbaijani, it is also useful to use the morphology information in word representations [1].

Sometimes related languages use different scripts but have similar pronunciation. In this case, we can convert the text data into their pronunciation representation using International Phonetic Alphabet (IPA). Epitran is a useful python package that transforms text into their IPA representation [6].

Recently, several approaches are proposed to improve the word segmentation for standard NMT models [7, 4]. It is possible that these improvements would be helpful for multilingual NMT as well.

5.4 Better Modeling

You can also improve the NMT architecture or optimization to better model data from multiple languages. [14] propose three strategies to improve one-to-many translation, including better initialization and language specific embedding. [16] propose adding language-aware modules in the NMT model.

5.5 And many others...

There are many potential directions for improving multilingual training. We encourage you to do more literature research, and even come up with your own method!

6 Grading

Please write up a report documenting the results using the ACL format. Here are the grading criterion:

- B: reproduce the bilingual and multilingual baselines in Table. 1 and Table. 2. To account for variance in experiment runs, your experiments are allowed to be 0.5 lower than our number.
- B+: analyze how multilingual training helps the low-resource languages, and clearly document your findings. You can analyze any phenomenon you find interesting. For example, does multilingual training perform better when translating from English to the low-resource language, or from English to the low-resource language? Why is that?
- A-: implement at least one pre-existing method, clearly document the results and analyze why it does or does not work.
- A/A+: implement several methods (this could be multiple pre-existing methods, or one pre-existing method and a novel method) and compare their performance, clearly document the results and analyze why they work or don't work.

You are allowed to use external corpora or linguistic resources. We won't grade the project based on the amount of improvement, so simply collecting and using large amounts of external data is not the only thing you should be doing (but if your collection methodology is particularly interesting this can be counted as a novel method – consult with TAs/Instructors about this if you wish). Instead, we will focus on evaluating whether a systematic comparison and analysis is clearly documented. You could compare these different strategies of using monolingual data, or explore what is the best way to combine them together.

7 Submission

Your submission consists of two parts: Code and a writeup. Put all your code in a folder named `code` including instructions on how to run the experiments. Rename the writeup as `writeup.pdf` and compress both on them as `assign2.tar.gz`. This file must be submitted to Canvas (link on the course website).

The assignment is due on October 12th, 11:59 pm ET.

References

[1] Chaudhary, A., Zhou, C., Levin, L., Neubig, G., Mortensen, D. R., and Carbonell, J. Adapting word embeddings to new languages with morphological and phonological subword representations. In *EMNLP* (2018).

[2] Edunov, S., Ott, M., Auli, M., and Grangier, D. Understanding back-translation at scale. In *EMNLP* (2018).

[3] He, J., Gu, J., Shen, J., and Ranzato, M. Revisiting self-training for neural sequence generation. In *ICLR* (2020).

[4] He, X., Haffari, G., and Norouzi, M. Dynamic programming encoding for subword segmentation in neural machine translation. In *ACL* (2020).

[5] Lin, Y., Chen, C., Lee, J., Li, Z., Zhang, Y., Xia, M., Rijhwani, S., He, J., Zhang, Z., Ma, X., Anastasopoulos, A., Littell, P., and Neubig, G. Choosing transfer languages for cross-lingual learning. In *ACL* (2019).

[6] Mortensen, D. R., Dalmia, S., and Littell, P. Epitran: Precision G2P for many languages. In *LREC* (May 2018).

[7] Provilkov, I., Emelianenko, D., and Voita, E. BPE-dropout: Simple and effective subword regularization. In *ACL* (2020).

[8] Qi, Y., Sachan, D. S., Felix, M., Padmanabhan, S., and Neubig, G. When and why are pre-trained word embeddings useful for neural machine translation? In *NAACL* (2018).

[9] Sennrich, R., Haddow, B., and Birch, A. Improving neural machine translation models with monolingual data. In *ACL* (2016).

[10] Siddhant, A., Bapna, A., Cao, Y., Firat, O., Chen, M., Kudugunta, S., Arivazhagan, N., and Wu, Y. Leveraging monolingual data with self-supervision for multilingual neural machine translation. In *ACL* (2020).

[11] Sun, J., Ahn, H. A., Park, C. Y., Tsvetkov, Y., and Mortensen, D. R. Ranking transfer languages with pragmatically-motivated features for multilingual sentiment analysis. In *Arxiv* (2020).

[12] Wang, X., and Neubig, G. Target conditioned sampling: Optimizing data selection for multilingual neural machine translation. In *ACL* (2019).

[13] Wang, X., Pham, H., Arthur, P., and Neubig, G. Multilingual neural machine translation with soft decoupled encoding. In *ICLR* (2019).

[14] Wang, Y., Zhang, J., Zhai, F., Xu, J., and Zong, C. Three strategies to improve one-to-many multilingual translation. In *ACL* (2018).

[15] Xia, M., Kong, X., Anastasopoulos, A., and Neubig, G. Generalized data augmentation for low-resource translation. In *ACL* (2019).

[16] Zhang, B., Williams, P., Titov, I., and Sennrich, R. Improving massively multilingual neural machine translation and zero-shot translation. In *ACL* (2020).