CS4248 Natural Language Processing
AY 2024/25 Semester 1

You can choose your final project from a list of three pre-defined topics:

- Grammatical error correction
- Extractive question answering
- Machine translation

I. Grammatical Error Correction

Grammatical error correction (GEC) deals with correcting writing errors in a written text in English. These errors include spelling errors, punctuation errors, grammar errors, word choice errors, etc. For this final project topic, your goal is to implement a state-of-the-art grammatical error correction system. Evaluation of your GEC system is to be carried out on the CoNLL-2014 and BEA-2019 shared tasks. Below are the papers describing the shared tasks:

https://aclanthology.org/W14-1701.pdf
https://aclanthology.org/W19-4406.pdf

Training data

You can obtain the training data from:
https://www.cl.cam.ac.uk/research/nl/bea2019st/

The training data comes from four sources:

FCE v2.1:
https://www.cl.cam.ac.uk/research/nl/bea2019st/data/fce_v2.1.bea19.tar.gz

Lang-8 Corpus of Learner English:
Request this corpus from:
https://docs.google.com/forms/d/e/1FAIpQLSflRX3h5QYxegivjHN7SJ194OxZ4XN_7Rt0cNpR2YbmNV-7Ag/viewform
An email with a download link will be sent to you.

NUCLE (NUS Corpus of Learner English):
You can obtain this corpus from the Canvas Project folder.

W&I+LOCNESS v2.1:
https://www.cl.cam.ac.uk/research/nl/bea2019st/data/wi+locness_v2.1.bea19.tar.gz

Pre-processing

Among the downloaded files, you can find the files that together constitute the training data:

```
fce/m2/fce.train.gold.bea19.m2
fce/m2/fce.dev.gold.bea19.m2
lang8/lang8.train.auto.bea19.m2
nucle/bea2019/nucle.train.gold.bea19.m2
wi+locness/m2/A.train.gold.bea19.m2
```

```
wi+locness/m2/B.train.gold.bea19.m2
wi+locness/m2/C.train.gold.bea19.m2
```

The above files are in the so-called M2 format. Essentially, each `.m2` file consists of blocks separated by a blank line between two blocks. Each block consists of one tokenized sentence, followed by annotations of the errors found in the sentence. Each annotated error includes the start token position, end token position, error type, and the replacement string. You can find detailed description of M2 format in the M2 scorer readme file.

You can use the Python script `m2_to_parallel.py` to convert an M2 file into two files: a source file and a target file, consisting of the source sentences (`.src`) written by English learners and target sentences (`.tgt`) which are the target or corrected sentences. The command to generate the corresponding source and target files from an M2 file is as follows:

```
python m2_to_parallel.py --data filename.m2 --erroneous_only
```

Using the `--erroneous_only` argument, only the erroneous portion of the data (i.e., where a corrected sentence differs from its source sentence) is used to generate the `.src` and `.tgt` files.

Development data

The file that contains the development data is:

```
wi+locness/m2/ABCN.dev.gold.bea19.m2
```

Source and target files are generated similarly from `ABCN.dev.gold.bea19.m2` using `m2_to_parallel.py`, but **without** the argument `--erroneous_only`.

Test data

The test data (including the source sentences and the gold-standard error annotations) of CoNLL-2014 shared task can be downloaded from:

https://www.comp.nus.edu.sg/~nlp/conll14st/conll14st-test-data.tar.gz

The test data of BEA-2019 shared task (only the source sentences but without the gold-standard error annotations) is the following file:

```
wi+locness/test/ABCN.test.bea19.orig
```

Scorers

The M2 scorer is used to score the output in the CoNLL-2014 shared task. The M2 scorer can be downloaded from:

https://www.comp.nus.edu.sg/~nlp/sw/m2scorer.tar.gz

The ERRANT scorer is used to score the output in the BEA-2019 shared task. The ERRANT scorer can be downloaded from:
https://github.com/chrisjbryant/errant

Since only the source sentences in the test set of BEA-2019 shared task are available, in order to score your output, you must upload the corrected sentences output by your system to CodaLab's server:

https://codalab.lisn.upsaclay.fr/competitions/4057

II. Extractive question answering

This task involves implementing a question answering system such that given a passage and a question in English, the system provides an answer to the question based on the passage, where the answer is a segment of text (called a span) from the passage.

Training and test data

We will use SQuAD (Stanford Question Answering Dataset) in this project for training and testing. The training data (train-v1.1.json) and test data (dev-v1.1.json) have been released on Canvas Project folder. Details of this task and the dataset can be found in the following paper:

SQuAD: 100,000+ Questions for Machine Comprehension of Text
Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang
Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing
https://aclanthology.org/D16-1264.pdf

Scorer

Evaluation of your implemented QA system is to be carried out using the official evaluation script evaluate-v2.0.py:

https://worksheets.codalab.org/rest/bundles/0x6b567e1cf2e041ec80d7098f031c5c9e/contents/blob/

Two evaluation metrics are used:

Exact match: This metric measures the percentage of your system's answers that *exactly* match any one of the gold-standard answers.

F1 score (macro-averaged): This metric measures the maximum F1 score based on the *overlap* between the system's answer and a gold-standard answer (treated as bags of tokens) for a question, and then averages the F1 scores over all questions.

III. Machine translation

This task involves implementing a machine translation (MT) system: Given a sentence in Chinese, the system provides a translation of that sentence into English.

Test data

There are two test sets, available on Canvas Project folder:

1. A simpler test set consisting of short sentences taken from the Tatoeba corpus.
2. A more complex test set consisting of sentences from the WMT-2022 general MT task.

Details of the WMT-2022 general MT task and its Chinese-to-English test set can be found in:

Tom Kocmi, Rachel Bawden, Ondřej Bojar, et al. 2022. Findings of the 2022 Conference on Machine Translation (WMT22). In Proceedings of the Seventh Conference on Machine Translation (WMT 2022). https://aclanthology.org/2022.wmt-1.1.pdf

The complete WMT-2022 test set can also be downloaded from:
https://github.com/wmt-conference/wmt22-news-systems/archive/refs/tags/v1.1.tar.gz

Training data

We also provide a training set of parallel Chinese-English sentence pairs on Canvas, which can also be downloaded from:
https://huggingface.co/datasets/haoranxu/ALMA-Human-Parallel

Refer to the following paper about this training set:

Haoran Xu, Young Jin Kim, Amr Sharaf, Hany Hassan Awadalla. 2024. A Paradigm Shift in Machine Translation: Boosting Translation Performance of Large Language Models. In Proceedings of the International Conference on Learning Representations (ICLR 2024).

Scorers

Your MT output will be evaluated using two evaluation metrics: BLEU and COMET.

BLEU (Papieni et al., 2002): This metric evaluates the quality of a translated text by calculating the precision of matching n-grams (up to 4-grams) between the system's translation and one or more reference translations, while applying a brevity penalty to penalize overly short translations. You can use the SacreBLEU library (Post, 2018) (source code: https://github.com/mjpost/sacrebleu) to compute the score.

Use the following command to compute the BLEU score using a reference file:
```
sacrebleu -tok 13a -w 2 {reference_file} < {prediction}
```

Use the following command to compute the BLEU score on the WMT-2022 test set:
```
sacrebleu -t wmt22 -l zh-en -m bleu -i ${model_output}
```

COMET (Rei et al., 2020): This metric measures the similarity between the sentence embeddings of the translated sentence and the reference translation using a transformer model trained on the human evaluation data from previous WMT shared task submissions. You can find the documentation of COMET at https://unbabel.github.io/COMET/html/index.html.

Use the following command to compute the COMET score:
```
comet-score -s ${source_file} -t ${prediction} -r ${reference_file} --batch_size 256 --model Unbabel/wmt22-comet-da --gpus 1
```

Refer to the following papers on machine translation evaluation:

Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002).

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In Proceedings of the Third Conference on Machine Translation (WMT 2018).

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A Neural Framework for MT Evaluation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020).