Naomi Bleiker (nbleiker, 17-648-981)
Alison Y. Kim (akim, 21-738-877)

Exercise 05

Maschinelle Übersetzung
2. Juni 2022

1 Experiments with byte pair encoding

We used **DE-NL** data `{dev, test, train}.de-nl.{de, nl}` for the experiments, with `.de` and `.nl` as the source and target files, respectively.

1.1 Submission

- `configs/*`: Configuration files for models A, B, and C
- `data/*`: Train, validation, and test corpora
- `models/transformer_{*}/{train, translate}.log`: Training and translation logs for models A, B, and C
- `scripts/`: These are to be run from the **main exercise directory**, `mt-2023-ex05`. (Example program call: `bash scripts/evaluate.sh`)
  - `evaluate.sh`: Evaluate models
  - `learn_bpe.sh`: Follow best practices[1] to learn BPE for Experiments B and C
  - `train.sh`: Train models
- `translations/transformer_{a, b, c}/test.*`: Translation hypotheses from each model

1.2 Training

The training script was modified to train all models of interest at once, i.e. the user may run the script only once for all models. This is useful for training multiple models consecutively, as it may be inconvenient to wait for an unspecified amount of time for one model to finish before readjusting the script and running it for the next model. Of course, the user may train a single model with the modified implementation.

All model configuration files are assumed to be prefixed by `transformer_`. If the user desires a different naming scheme and/or wishes to train non-transformer models, **line 22** should be modified. Otherwise, the user can directly modify the variable `model_ext` (**line 19**).

In addition to defining the model name, the user should modify the following variables according to the particular system they are training:

- `src`: Source language (2-letter ISO code)
- `trg`: Target language (2-letter ISO code)
- `num_threads`: Number of threads in CUDA block, if CUDA is available
- `device`: Number of GPU devices

[1] https://github.com/rsennrich/subword-nmt/tree/master#best-practice-advice-for-byte-pair-encoding-in-nmt

After making these adjustments, the user should activate the virtual environment if not already done. Finally, the script should be run from the **main exercise directory** as such:

```
(torch3) USER/../mt-2023-ex05 % bash scripts/train.sh
```

1.2.1 Models B and C: BPE

Before training models B and C, we had to learn a subword vocabulary. For this, we used the package `subword-nmt`, which should be installed before running the BPE script.

Following the recommendation on page 4 of the worksheet, we built a joint vocabulary file for the two languages and removed the vocabulary counts.

The script can be run from the **main exercise directory** as such:

```
(torch3) USER/../mt-2023-ex05 % pip install subword-nmt # if not already done
(torch3) USER/../mt-2023-ex05 % bash scripts/learn_bpe.sh
```

1.3 Evaluation

For convenience, the evaluation script was modified in a similar fashion to that of the training script. The same five variables are to be defined according to the user's needs. Then, the evaluation script can be run from the **main exercise directory** with:

```
(torch3) USER/../mt-2023-ex05 % bash scripts/evaluate.sh
```

For Part 1, we chose a beam size of $k = 1$, i.e. greedy decoding, so that we can explore the effects of $k > 1$ for Part 2.

1.3.1 Automatic: BLEU

Table 1 displays the BLEU scores of the automatic translations of each system. Experiments A and B differed in the vocabulary on which the model was trained. From the noticeable jump in BLEU score between these two experiments, we can conclude that BPE improves translation quality for the DE $\rightarrow$ NL direction.

Experiments B and C both used BPE, differing only in vocabulary size, respectively 2,000 and 50,000. There is a slight increase in the BLEU score from model B to C, suggesting that an increased vocabulary size improves translation quality, at least up to a certain point. We chose `num_merges = 2,000` for both training experiments, as a larger value might not have resulted in learning a useful representation of relatively rarely occurring subwords, given the relatively small dataset size (100,000 training sentences) [4]. We expect that increasing the value may have only a *marginally* positive

effect on the translation quality, as measured by a higher BLEU score.

The BLEU scores themselves are low and indicate relatively poor model performance [2]. From a manual inspection, we noticed a significant amount of `<unk>` tokens in Model A's translations, which makes sense given the limited vocabulary size of 2000. Interestingly, Model B's translations contained no `<unk>` tokens, while C's translations did when preceding an apostrophe or punctuation. We wondered whether the presence of `<unk> quot;` and `<unk> apos;` tokens affected the BLEU score. Thus, we manually replaced these with `''/''` and `'`, respectively,[2] and recalculated the BLEU score. As expected, the BLEU score increased (see row "Experiment *C, cleaned*" in Table 1).

Experiment	BPE	Vocabulary size	BLEU
A	No	2000	11.3
B	Yes	2000	16.9
C	Yes	50000	17.8

Table 1: BLEU scores.

1.3.2 Manual: Human

To qualitatively assess the translations, we recruited an L1 Dutch/B2 German speaker.[3] He was shown a total of six sentences (two per model).

Two source sentences (DE) were randomly selected from the test file `data/test.de-nl.nl`:

- **Source 1 (DE, lines 1105-1106)**: Denn an einem bestimmten Punkt musste ich mir sagen: "Was mache ich eigentlich hier? Wieso mache ich das? Was kommt dabei heraus?"
- **Source 2 (DE, line 1328)**: In den letzten Jahren haben wir also angefangen, über das Glück der beiden Arten des Selbst zu lernen.

The corresponding hypotheses (NL) were selected from each model's respective hypothesis file, `translations/transformer_{a, b, c}/test.*`.

[2]The cleaned sentences were saved to `translations/transformer_c/test.transformer_c_clean.nl`.
[3]Bedankt, Jeroen De Vrieze!

Model	Source (DE)	Hypothesis (NL)	Qualitative Evaluation
A	1	Want op een bepaald punt moest ik zeggen: "Wat doe ik hier? Waarom doe ik data Wat komt?"	"Op een bepaald punt" is ok. But in this type of philosophical questioning, I'd opt for the phrasing "Op een gegeven moment".
A	2	In de afgelopen jaren zijn we begonnen te leren over het geluk van de twee soorten van het zelf te leren.	In de afgelopen jaren / in de laatste jaren : both alright. They translated "lernen" twice making the verb structure a bit weird. But you can put the infinitive "te leren" almost anywhere in the sentence. [We Dutch speakers are] flexible. Might put it a little earlier in the sentence to break the sentence and make it easier to follow.
B	1	Want aan een bepaalde punt moest ik zeggen: "Wat doe ik hier? Waarom doe ik dit?"	"Waarom doe ik dat". Again not wrong but I'd use 'dit' instead of 'dat' not knowing the context and assuming the author is questioning his/her life path.
B	2	In de afgelopen jaren hebben we het over het geluk van de twee soorten van het zelf te leren.	"Het zelf : literal translation is fine. Again not knowing the context, might opt for something else or maybe alternatively using a capital for 'Zelf'. The verb "anfangen" is not translated. The sentence doesn't make sense because of it.
C	1	Want op een bepaald punt moest ik zeggen: Wat doe ik hier hier? Waarom doe ik dit?	'Dat' is used for something concrete, almost tangible, something you can point to and say 'dat daar'. 'dit' is more distant. Last sentence is most difficult. "Was kommt dabei heraus" was kept out of translations 2 and 3 and in 1 it was an incomplete sentence. Again not knowing what *dabei* refers to and assuming it's rhetorical, I'd translate to sth like "wat brengt me dit / wat levert me dit op?" But you can go a lot more liberal in the translation of course
C	2	In de laatste jaren begonnen we te leren over het geluk van beide soorten.	"zelf" is not translated

Table 2: Qualitative evaluations from an L1 Dutch/B2 German speaker.

2 Impact of beam size on translation quality

2.1 Submission

- `scripts/`: These are to be run from the **main exercise directory**, `mt-2023-ex05`. Example program call: `bash scripts/translate.sh`.
  - `graph.py`: Graph BLEU scores and translation times
  - `modify_k.py`: Modify the beam size value in a selected model's YAML configuration file
  - `translate.sh`: Produce translations using a specific model(s) for varying beam sizes
- `translations/graphs/*`: Graphical representations of translation quality metrics
- `translations/logs/*`: BLEU scores and times for each beam size experiment
- `translations/transformer_c/hyps*`: Translation hypotheses from model C for beam sizes $k \in [2, 3, 4, 5, 6, 7, 8, 10, 12, 15, 20]$

2.2 Results

Beam Size k	**Time (sec)**	**BLEU**
1 (greedy, from Part 1)	92	17.8
2	40	17.9
3	52	17.8
4	65	17.9
5	87	17.8
6	103	17.8
7	124	17.7
8	141	17.5
10	168	17.3
12	201	17.2
15	260	17.2
20	350	17.0

Table 3: Inference metrics as a function of beam size.

2.3 Discussion

We were expecting a fairly unimodal distribution of BLEU scores centered around a specific beam size, with an obvious peak at a particular beam size value. This is clearly not the case according to the values we obtained. We believe that the model's overall low performance can be attributed to the small dataset size, small number of merges, and the small vocabulary size, among other factors.

We were *not* expecting a monotonic increase in text quality, at least measured by a BLEU score, as too high a beam size has been shown to degrade the quality of generated text [3], [1], [5]. The same literature, among others, has established $k = 5$ as an upper limit for producing high-quality

text, which is more or less supported by Table 3 for this experiment (BLEU score begins to decline monotonically from $k = 7$). For future experiments, we might stick to $k = 5$ and test out other beam sizes that are slightly larger and slightly smaller than 5, such as a range from $[4, 7]$.

References

[1] Eldan Cohen and Christopher Beck. "Empirical Analysis of Beam Search Performance Degradation in Neural Sequence Models". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 1290–1299. URL: https://proceedings.mlr.press/v97/cohen19a.html.

[2] *Evaluating models: Understanding the BLEU score*. https://cloud.google.com/translate/automl/docs/evaluate#bleu. Accessed: 2023-05-40.

[3] Philipp Koehn and Rebecca Knowles. *Six Challenges for Neural Machine Translation*. 2017. arXiv: 1706.03872 [cs.CL].

[4] *Number of merge operations*. https://github.com/rsennrich/subword-nmt/issues/61. Accessed: 2023-05-40.

[5] Yilin Yang, Liang Huang, and Mingbo Ma. "Breaking the Beam Search Curse: A Study of (Re-)Scoring Methods and Stopping Criteria for Neural Machine Translation". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3054–3059. DOI: 10.18653/v1/D18-1342. URL: https://aclanthology.org/D18-1342.