Joey NMT v2.0
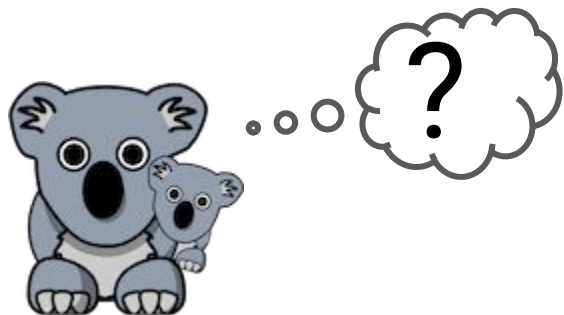# Code Quiz

Answer these questions to explore Joey NMT!

1. Training

You have successfully installed Joey NMT and written a configuration file `config.yaml`.

Which command would you use to start training a model with this configuration?

1. Training

You have successfully installed Joey NMT and written a configuration file `config.yaml`.

Which command would you use to start training a model with this configuration?

```
python -m joeynmt train config.yaml
```

2. Saving

How do you know your model was saved during training?

☐ Check in the validation report whether there's any line ending with "*".

☐ Check the training log if it says it saved checkpoints.

☐ Check if there are any `*.ckpt` files in the model directory.

☐ The model always gets saved during training.

2. Saving

How do you know your model was saved during training?

———————————————————————

☑️ Check in the validation report whether there's any line ending with "*".

☑️ Check the training log if it says it saved checkpoints.

☑️ Check if there are any `*.ckpt` files in the model directory.

☐ The model always gets saved during training.

3. Testing

When using Joey NMT in test mode, can you specify the checkpoint for testing anywhere outside the configuration file?

☐ Yes

☐ No

3. Testing

When using Joey NMT in test mode, can you specify the checkpoint for testing anywhere outside the configuration file?

☑️ Yes

☐ No

```
$ python -m joeynmt test configs/small.yaml --ckpt model_dir/avg.ckpt
```

4. Parameters

How many parameters does the model specified in `configs/transformer_small.yaml` have in total?

This includes all parameter weights and biases of the model, including e.g. the embeddings. Hint: Joey NMT computes it for you.
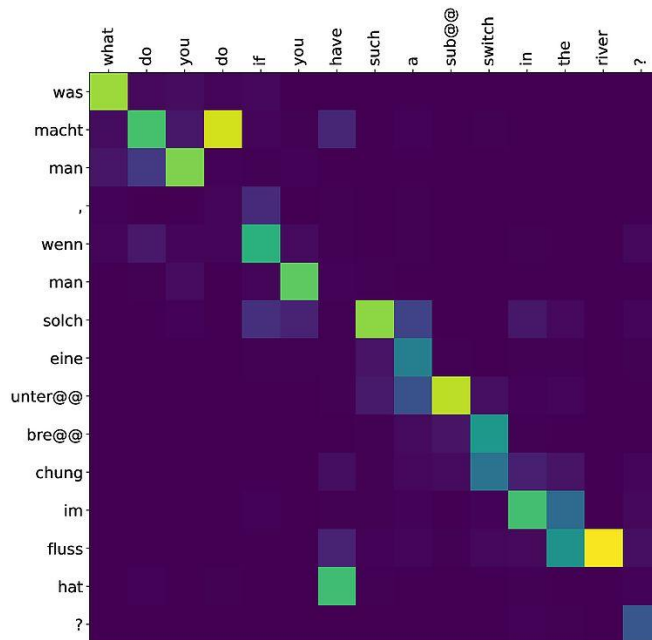
4. Parameters

How many parameters does the model specified in `configs/transformer_small.yaml` have in total?

This includes all parameter weights and biases of the model, including e.g. the embeddings. Hint: Joey NMT computes it for you.
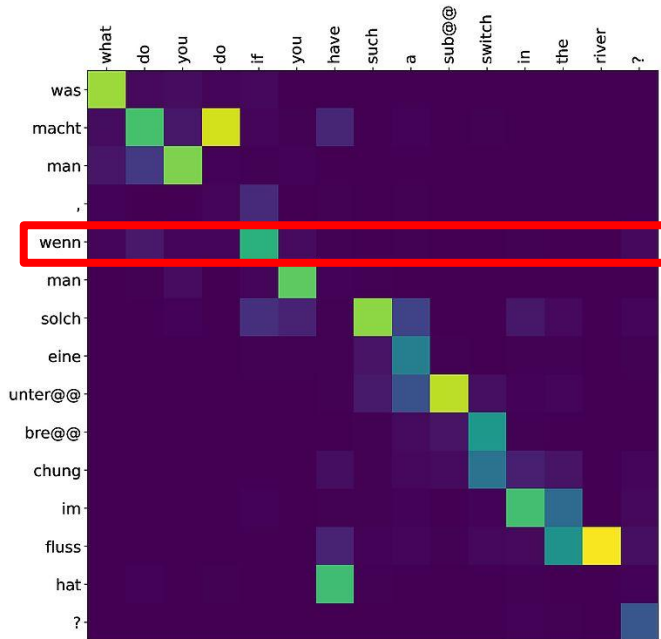

277,504

5. Attention

Which source token receives most attention when generating the target word "if"?

5. Attention

Which source token receives most attention when generating the target word "if"?

6. Speed

How do you find out how fast your model trains (including validations)?

6. Speed

How do you find out how fast your model trains (including validations)?

The number of tokens per second is logged and reported in the log file.

7. Pre-processing

What kind of pre-processing does Joey NMT v2 do for you? (if specified)

☐ splitting into sub-word units (BPEs)
☐ data filtering by source/target length ratio
☐ data filtering by source and target length
☐ tokenization
☐ lowercasing

7. Pre-processing

What kind of pre-processing does Joey NMT do for you? (if specified)

_____

☑️ splitting into sub-word units (BPEs)

☐ data filtering by source/target length ratio

☑️ data filtering by source and target length

☑️ tokenization

☑️ lowercasing

https://github.com/joeynmt/joeynmt/blob/main/joeynmt/tokenizers.py#L95
https://github.com/joeynmt/joeynmt/blob/main/joeynmt/tokenizers.py#L62

8. Post-processing

What kind of post-processing does Joey NMT v2 do for you? (if specified)

☐ recasing
☐ detokenization
☐ subword merging ("un-BPE-ing")
☐ delemmatization

8. Post-processing

What kind of post-processing does Joey NMT do for you? (if specified)

☐ recasing
☑ de-tokenization
☑ subword merging ("un-BPE-ing")
☐ de-lemmatization

https://github.com/joeynmt/joeynmt/blob/main/joeynmt/tokenizers.py#L108

9. Checkpoints

In a debugging scenario, you don't want to store the best checkpoints for your current model. There's a line that you can write in your configuration file to make the model not save these checkpoints during training. What is this line?

9. Checkpoints

In a debugging scenario, you don't want to store the best checkpoints for your current model. There's a line that you can write in your configuration file to make the model not save these checkpoints during training. What is this line?

```
keep_best_ckpts: 0
```

Joey NMT always saves the latest checkpoint.
If you set `keep_best_ckpts` to -1, all checkpoints will be saved.
If you set it to n>0, only the n best will be saved.

10. Model size

Change the following model configuration to use three encoder layers.

```
encoder:
    rnn_type: "lstm"
    embeddings:
        embedding_dim: 16
        hidden_size: 64
        bidirectional: True
```

Which line would you have to add?

10. Model size

Change the following model configuration to use three encoder layers.

```
encoder:
    rnn_type: "lstm"
    embeddings:
        embedding_dim: 16
        hidden_size: 64
        bidirectional: True
```

Which line would you have to add?

_____

```
    num_layers: 3
```

11. Data path

Which line would you have to add to the data configuration below to use my
`home/my_dir/my_data.en` as test input file? Hint: mind the file ending.

```
data:

    train: "test/data/reverse/train"
    dev: "test/data/reverse/dev"
    level: "word"
    dataset_type: "plain"
    src:
        lang: "en"
    trg:
        lang: "fr"
```

11. Data path

Which line would you have to add to the data configuration below to use my `home/my_dir/my_data.en` as test input file? Hint: mind the file ending.

```
data:

    train: "test/data/reverse/train"
    dev: "test/data/reverse/dev"
    level: "word"
    dataset_type: "plain"
    src:
        lang: "en"
    trg:
        lang: "fr"

  test: "my_home/my_dir/my_data"
```

12. Training hyperparameters

Modify the following config such that it uses a constant learning rate of 0.02.

```
training:
    optimizer: "adam"
    learning_rate: 0.001
    clip_grad_norm: 1.0
    batch_size: 10
    scheduling: "plateau"
    patience: 5
    decrease_factor: 0.5
    early_stopping_metric: "bleu"
    epochs: 6
    validation_freq: 1000
    logging_freq: 100
    model_dir: "reverse_model"
    max_output_length: 30
```

12. Training hyperparameters

Modify the following config such that it uses a constant learning rate of 0.02.

```
training:
    optimizer: "adam"
    learning_rate: 0.02
    clip_grad_norm: 1.0
    batch_size: 10
    scheduling: "plateau"
    patience: 5
    decrease_factor: 0.5
    early_stopping_metric: "bleu"
    epochs: 6
    validation_freq: 1000
    logging_freq: 100
    model_dir: "reverse_model"
    max_output_length: 30
```

13. Vocabulary generation

When the vocabulary is extracted from training data, we keep only the `src:voc_limit` / `trg:voc_limit` most frequent tokens that occur at least `src:min_freq` / `trg:min_freq` times in the training data.

For our example, the vocabulary limit is 15, while the minimum frequency is 3. After counting the tokens in the training data and filtering by minimum frequency, we have the following counts:

i: 22   you: 14   and: 9   ,: 9   to: 7   if: 6   joey: 5   't: 5   anymore: 5   're: 4
scared: 4   be: 4   angry: 4   but: 3   it: 3   out: 3   don: 3   get: 3   oh: 3   the: 3

Which of those tokens would not end up in the vocabulary, according to Joey NMT's vocabulary building?

- ☐ it
- ☐ the
- ☐ oh
- ☐ get
- ☐ don
- ☐ but
- ☐ out

13. Vocabulary generation

When the vocabulary is extracted from training data, we keep only the `src_voc_limit` / `trg_voc_limit` most frequent tokens that occur at least `src_min_freq` / `trg_min_freq` times in the training data.

For our example, the vocabulary limit is 15, while the minimum frequency is 3. After counting the tokens in the training data and filtering by minimum frequency, we have the following counts:

i: 22	you: 14	and: 9	,: 9	to: 7	if: 6	joey: 5	't: 5	anymore: 5	're: 4
scared: 4	be: 4	angry: 4	but: 3	it: 3	out: 3	don: 3	get: 3	oh: 3	the: 3

Which of those tokens would not end up in the vocabulary, according to Joey NMT 's vocabulary building?

☑ it　　☑ the　　☑ oh　☑ get

☐ don　☐ but　☑ out

14. Special tokens

Which is the default token used for marking the end-of-sequence position in Joey NMT ?

e.g. <end> or [EOS]?

14. Special tokens

Which is the default token used for marking the end-of-sequence position in Joey NMT ?

e.g. <end> or [EOS]?

</s>

https://github.com/joeynmt/joeynmt/blob/main/joeynmt/constants.py#L9

15. Data iterators

Training and validation data are treated differently in Joey NMT - but in which ways?

For example, if you choose "sampling", it means that validation and training data are handled differently with respect to how batches are sampled from it.

☐ Shuffling
☐ Filtering
☐ Subword segmentation
☐ Embedding
☐ Sampling

15. Data iterators

Training and validation data are treated differently in Joey NMT - but in which ways?

For example, if you choose "sampling", it means that validation and training data are handled differently with respect to how batches are sampled from it.

☑️ Shuffling
☑️ Filtering
☐ Subword segmentation
☐ Embedding
☑️ Sampling

https://github.com/joeynmt/joeynmt/blob/main/joeynmt/data.py#L239 shuffle & sample

https://github.com/joeynmt/joeynmt/blob/main/joeynmt/tokenizers.py#L91 length filtering

16. Training loop

Where is the training for-loop over epochs defined? Find the line in the code.

16. Training loop

Where is the training for-loop over epochs defined? Find the line in the code.

```
for epoch_no in range(self.epochs):
```

https://github.com/joeynmt/joeynmt/blob/main/joeynmt/training.py#L446

17. End of training

When does training end? (Assuming there are no technical problems like memory errors etc.)

We refer to settings in the configuration file, e.g. learning rate.

☐ When the minimum learning rate (`learning_rate_min`) has been reached.

☐ When the maximum validation scores has been reached.

☐ Just after `keep_best_ckpts` checkpoints have been saved.

☐ When Joey NMT gets tired.

☐ When all epochs have been completed.

☐ When you interrupt the training process with Ctrl+C.

17. End of training

When does training end? (Assuming there are no technical problems like memory errors etc.)

We refer to settings in the configuration file, e.g. learning rate.

✅ When the minimum learning rate (`learning_rate_min`) has been reached.
☐ When the maximum validation scores has been reached.
☐ Just after `keep_last_ckpts` checkpoints have been saved.
☐ When Joey NMT gets tired.
☑ When all epochs have been completed.
☑ When you interrupt the training process with Ctrl+C.

18. Model

What does `model.forward(return_type="decode")` return?

18. Model

What does `model.forward(return_type="decode")` return?

- output vectors
- decoder hidden states
- attention probabilities
- attention vectors (only for RNN, else None)

Model: https://github.com/joeynmt/joeynmt/blob/main/joeynmt/model.py#L127
RNN: https://github.com/joeynmt/joeynmt/blob/main/joeynmt/decoders.py#L341
Transformer: https://github.com/joeynmt/joeynmt/blob/main/joeynmt/decoders.py#L560

19. Initialization

How are forget gates of LSTMs initialized by default?

☐ All ones
☐ Random normal initialization
☐ Random uniform initialization
☐ All zeros
☐ Xavier initialization

19. Initialization

How are forget gates of LSTMs initialized by default?

☑ All ones
☐ Random normal initialization
☐ Random uniform initialization
☐ All zeros
☐ Xavier initialization

https://github.com/joeynmt/joeynmt/blob/main/joeynmt/initialization.py#L31

20. Embeddings

In the configuration we can "freeze" the embeddings, so that they are not (further) trained:

```
embeddings:
    embedding_dim: 16
    freeze: True
```

Where does the freezing happen in JoeyNMT's code?

20. Embeddings

In the configuration we can "freeze" the embeddings, so that they are not (further) trained:

```
embeddings:
    embedding_dim: 16
    freeze: True
```

Where does the freezing happen in JoeyNMT's code?

_____

`freeze_params` in
https://github.com/joeynmt/joeynmt/blob/main/joeynmt/helpers.py#L567

21. Bidirectional

How are forward and backward states combined for a bidirectional recurrent encoder? Choose the correct tensor operation:

- ☐ `torch.add`
- ☐ `torch.cat`
- ☐ `torch.addbmm`
- ☐ `torch.sum`
- ☐ `torch.mul`
- ☐ `torch.pow`

21. Bidirectional

How are forward and backward states combined for a bidirectional recurrent encoder? Choose the correct tensor operation:

- ☐ `torch.add`
- ☑ `torch.cat`
- ☐ `torch.addbmm`
- ☐ `torch.sum`
- ☐ `torch.mul`
- ☐ `torch.pow`

22. Loss computation

Find the place where the batch loss is computed (comparing model outputs with targets), e.g.

```
train_batch_loss = my_loss_function(outputs, targets)
```

22. Loss computation

Find the place where the batch loss is computed (comparing model outputs with targets), e.g.
```
train_batch_loss = my_loss_function(outputs, targets)
```

```
batch_loss = self.loss_function(log_probs, **kwargs)
```

https://github.com/joeynmt/joeynmt/blob/main/joeynmt/model.py#L103

23. Batch

During training, the Batch object in JoeyNMT holds the reference sequence in `trg` for computing the loss and in `trg_input` for feeding it into the decoder.

What's the difference between those two tensors? (`batch.trg` vs. `batch.trg_input`)

☐ <s> is prepended to the first, otherwise no difference
☐ </s> is appended to the first and <s> is prepended to the second
☐ <s> is prepended to the second, otherwise no difference
☐ <s> is appended to the first and </s> is prepended to the second
☐ </s> is appended to the first, otherwise no difference

23. Batch

During training, the Batch object in JoeyNMT holds the reference sequence in trg for computing the loss and in trg input for feeding it into the decoder.

What's the difference between those two tensors? (`batch.trg` vs. `batch.trg_input`)

☐ <s> is prepended to the first, otherwise no difference
✅ </s> is appended to the first and <s> is prepended to the second
☐ <s> is prepended to the second, otherwise no difference
☐ <s> is appended to the first and </s> is prepended to the second
☐ </s> is appended to the first, otherwise no difference

https://github.com/joeynmt/joeynmt/blob/main/joeynmt/batch.py#L65
https://github.com/joeynmt/joeynmt/blob/main/joeynmt/data.py#L109

24. Inference algorithm

Where in the code is the decision made whether to decode greedily or with beam search?

Hint: it's an if-statement.

24. Inference algorithm

Where in the code is the decision made whether to decode greedily or with beam search?

Hint: it's an if-statement.

```
if beam_size < 2:
```

https://github.com/joeynmt/joeynmt/blob/main/joeynmt/search.py#L694

25. Validation score computation

Find the place where the validation score (here BLEU, `eval metric: bleu`) is computed.

25. Validation score computation

Find the place where the validation score (here BLEU, `eval metric: bleu`) is computed.

```
valid_scores[eval_metric] = bleu(
                valid_hyp_1best,
                valid_ref,  # detokenized ref
                **sacrebleu_cfg,
            )
```

https://github.com/joeynmt/joeynmt/blob/main/joeynmt/prediction.py#L249

26. BLEU computation

Which library is used for BLEU score computation?

26. BLEU computation

Which library is used for BLEU score computation?

[sacreBLEU](#)

27. Optimizers

Let's say you have invented a new optimizer and implemented it in PyTorch as `torch.optim.MagicOptimizer`. Now you want to use it in JoeyNMT by setting optimizer: magic in the configuration file. Which of JoeyNMT's Python files would you have to add the following lines to?

```
elif optimizer_name == "magic": # new awesome optimizer
    optimizer=torch.optim.MagicOptimizer(
        parameters, weight_decay=weight_decay,
        lr=learning_rate)
```

27. Optimizers

Let's say you have invented a new optimizer and implemented it in PyTorch as `torch.optim.MagicOptimizer`. Now you want to use it in JoeyNMT by setting optimizer: magic in the configuration file. Which of JoeyNMT's Python files would you have to add the following lines to?

```
elif optimizer_name == "magic": # new awesome optimizer
    optimizer=torch.optim.MagicOptimizer(
        parameters, weight_decay=weight_decay,
        lr=learning_rate)
```

```
builders.py
```

All done!

How many answers did you get right?

- [0](#)
- [>0 but <10](#)
- [>=10 but <20](#)
- [20-27](#)
- [>27](#)

0

Hmm, did you even try? :o

<10

Not bad for a start, but perhaps you can take more time for finding the answers next time. Try getting the most of the github search bar!

<20

That's well done! Some questions are really tricky, we know...

Here's a not quite biologically correct high-five from Joey:

20-27

Impressive! You researched well and dug deep into the code.

Joey is quite impressed, invites you over for a coffee, and offers you a leaf snack:

> 27

Redo the counting. Either you made a little mistake there, or you're very convinced of yourself - or both?