# ADL HW2 report

## Q1: Data processing

1. Tokenizer
    - Describe in detail about the tokenization algorithm you use. You need to explain what it does in your own ways.
        - **WordPiece (hfl/chinese-roberta-wwm-ext-large)**

          The vocabulary is initialized with each characters, and the most frequent combinations of symbols are iteratively added to the vocabulary.

          Note: The roberta-wwm-ext model is a roberta-like BERT, so the tokenizer algorithm is differenet from the original roberta (byte-level BPE).
          (source: https://arxiv.org/pdf/1906.08101.pdf)
2. Answer Span
    - How did you convert the answer span start/end position on characters to position on tokens after BERT tokenization?
        - **Offset mapping** returns the character offset (start_char, end_char), so we can use it to map to the tokenized start/end positions.

          (source: Tokenizer (huggingface.co))
    - After your model predicts the probability of answer span start/end position, what rules did you apply to determine the final start/end position?
        - Choose the position with the **highest probability** of start and end position.

## Q2: Modeling with BERTs and their variants

1. Describe
    - your model (configuration of the transformer model)
        - Context Selection

```
{
  "_name_or_path": "hfl/chinese-roberta-wwm-ext",
  "architectures": [
    "BertForMultipleChoice"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "classifier_dropout": null,
  "directionality": "bidi",
  "eos_token_id": 2,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "output_past": true,
  "pad_token_id": 0,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "position_embedding_type": "absolute",
  "torch_dtype": "float32",
  "transformers_version": "4.17.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```

- QA

```
{
  "_name_or_path": "hfl/chinese-roberta-wwm-ext-large",
  "architectures": [
    "BertForQuestionAnswering"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "classifier_dropout": null,
  "directionality": "bidi",
  "eos_token_id": 2,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 1024,
  "initializer_range": 0.02,
  "intermediate_size": 4096,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 16,
  "num_hidden_layers": 24,
  "output_past": true,
  "pad_token_id": 0,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "position_embedding_type": "absolute",
  "torch_dtype": "float32",
  "transformers_version": "4.17.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```

- performance of your model.

  | CS (on dev set) | QA (on dev set) | CS+QA (public score) | CS+QA (private score) |
  |---|---|---|---|
  | 0.96 | 0.823 | 0.78571 | 0.78590 |

  (CS: Context Selection)

- the loss function you used.

  **CrossEntropy()**

- The optimization algorithm (e.g. Adam), learning rate and batch size.

  |  | Optimization Algorithm | learning rate | batch size | accumulation step |
  |---|---|---|---|---|
  | CS | AdamW | 2e-5 | 5 | 10 |
  | QA | AdamW | 5e-5 | 2 | 5 |

2. Try another type of pretrained model and describe (QA)

   a. your model

```
{
  "_name_or_path": "bert-base-chinese",
  "architectures": [
    "BertForQuestionAnswering"
  ],
  "attention_probs_dropout_prob": 0.1,
  "classifier_dropout": null,
  "directionality": "bidi",
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 0,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "position_embedding_type": "absolute",
  "torch_dtype": "float32",
  "transformers_version": "4.17.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```

b. performance of your model (on dev set)

| | bert-base-chinese | chinese-roberta-wwm-ext-large |
|---|---|---|
| EM | 0.782 | **0.823** |

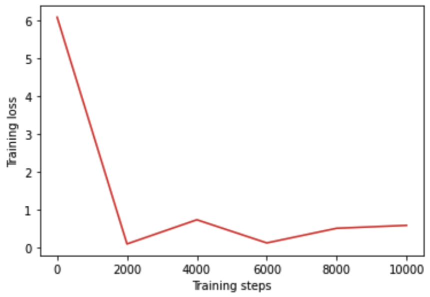
c. the difference between pretrained model (architecture, pretraining loss, etc.)

| | bert-base-chinese | chinese-roberta-wwm-ext-large |
|---|---|---|
| hidden size | 768 | 1024 |
| intermediate_size | 3072 | 4096 |
| num_attention_heads | 12 | 16 |
| num_hidden_layers | 12 | 24 |

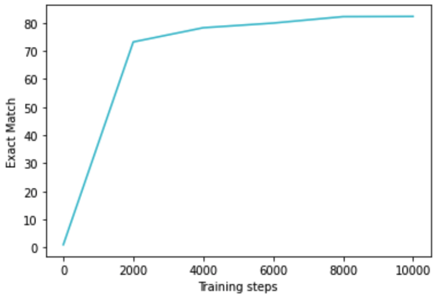
## Q3: Curves

1. Plot the learning curve of your QA model

    a. Learning curve of loss



    b. Learning curve of EM



## Q4: Pretrained vs Not Pretrained

1. Train a transformer model from scratch (without pretrained weights) on the dataset (you can choose either MC or QA)

2. Describe

    a. The configuration of the model and how do you train this model

```
{
  "_name_or_path": "hfl/chinese-roberta-wwm-ext",
  "architectures": [
    "BertForQuestionAnswering"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "classifier_dropout": null,
  "directionality": "bidi",
  "eos_token_id": 2,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 256,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 2,
  "num_hidden_layers": 3,
  "output_past": true,
  "pad_token_id": 0,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "position_embedding_type": "absolute",
  "torch_dtype": "float32",
  "transformers_version": "4.17.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```

I only load the configuration of the model, and reduce the model size by changing hidden_size from 768 to 256, num_attention_heads from 12 to 2, num_hidden_layers from 12 to 3.

b. the performance of this model v.s. BERT

| | w/o pretrained weights | w/ pretrained weights |
|---|---|---|
| EM | 5.2 | **82.3** |

## Q5: Bonus: HW1 with BERTs

- Train a BERT-based model on HW1 dataset and describe

1. your model

   a. Intent classification

```
{
  "_name_or_path": "roberta-base",
  "architectures": [
    "RobertaForSequenceClassification"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "classifier_dropout": null,
  "eos_token_id": 2,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-05,
  "max_position_embeddings": 514,
  "model_type": "roberta",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 1,
  "position_embedding_type": "absolute",
  "problem_type": "single_label_classification",
  "torch_dtype": "float32",
  "transformers_version": "4.17.0",
  "type_vocab_size": 1,
  "use_cache": true,
  "vocab_size": 50265
}
```

### b. Slot tagging

```
{
  "_name_or_path": "roberta-base",
  "architectures": [
    "RobertaForTokenClassification"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "classifier_dropout": null,
  "eos_token_id": 2,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-05,
  "max_position_embeddings": 514,
  "model_type": "roberta",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 1,
  "position_embedding_type": "absolute",
  "torch_dtype": "float32",
  "transformers_version": "4.17.0",
  "type_vocab_size": 1,
  "use_cache": true,
  "vocab_size": 50265
}
```

2. performance of your model.

    a. Intent classification

    | Bi-LSTM | Roberta |
    |---|---|
    | 0.916 | **0.961** |

    b. Slot tagging

    | Bi-LSTM | Roberta |
    |---|---|
    | 0.83 | **0.86** |

3. the loss function you used.

    Use **CrossEntropy()** in both tasks.

4. The optimization algorithm (e.g. Adam), learning rate and batch size.

| Optimization Algorithm | learning rate | batch size | epoch |
|---|---|---|---|
| AdamW | 3e-05 | 8 | 2 |