ADL_HW2

Q1: Data processing

1. Tokenizer

tokenization algorithm :

- wordpiece (Character-based)

detail:

1. Initialize the language model by base characters.
2. set the target size of language model and threshold of likelihood
3. Generate a new world unit by combining two units in the model. The new world unit increeases the likelihood the most of the data among all the posible combination.
4. Repeat step 3 until reaching the target size or likelihood being less than threshold.

2. Answer Span

Building a map to store a token's start and end corresponding to original data.

1. For both start and end, trace trough the map to find the match token. Time complexity is O(n).
2. Suppose model predicts the answer start from the 100th token to the 105th token. Just look up the map for start of the 100th token and end of the 105th token. Time complexity is O(1).

Q2: Modeling with BERTs and their variants

1. Describe (Default as TA)

- **bert-base-chinese**
    - **Loss function** : Cross Entrophy
    - **optimization algorithm** : AdamW
    - **learning rate** : 3e-5
    - **batch size**: 2 (**per_gpu_train_batch_size** : 1, **gradient_accumulation_steps** : 2)
    - **config**

```
{
  "architectures": [
    "BertForMultipleChoice" //"BertForQuestionAnswering" for QA
  ],
  "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
}
```

- **preformance**
    - **bert-base-chinese -> bert-base-chinese** : 0.72423 (Public Score)

2. Others type of model (with best preformance)

- **chinese-macbert-large**
    - **Loss function** : Cross Entrophy
    - **optimization algorithm** : AdamW
    - **learning rate** : 3e-5
    - **batch size**: 2 (**per_gpu_train_batch_size** : 1, **gradient_accumulation_steps** : 2)
    - **config** :

    ```
    {
      "architectures": [
        "BertForMaskedLM"
      ],
      "attention_probs_dropout_prob": 0.1,
      "directionality": "bidi",
      "gradient_checkpointing": false,
      "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,
      "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",
      "type_vocab_size": 2,
      "vocab_size": 21128
    }
    ```

- **chinese-roberta-wwm-ext-large**
  - **Loss function** : Cross Entrophy
  - **optimization algorithm** : AdamW
  - **learning rate** : 3e-5
  - **batch size**: 2
    - **per_gpu_train_batch_size** : 1
    - **gradient_accumulation_steps** : 2
  - **config** :

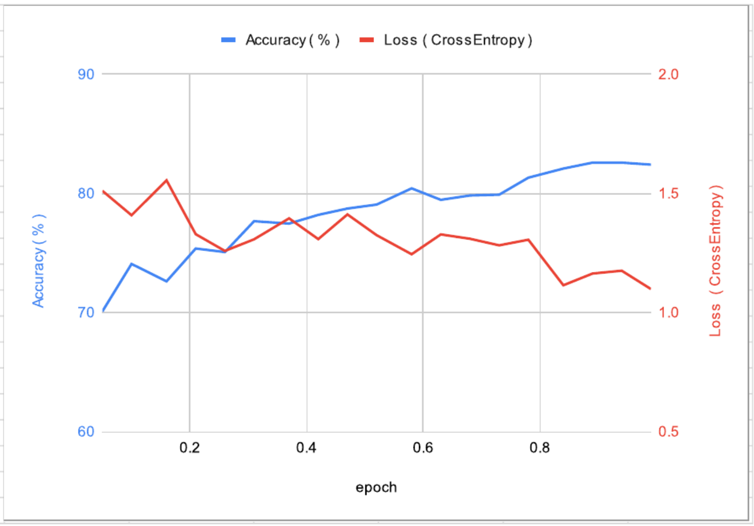
    ```
    {
      "architectures": [
        "BertForMaskedLM"
      ],
      "attention_probs_dropout_prob": 0.1,
      "bos_token_id": 0,
      "directionality": "bidi",
      "eos_token_id": 2,
      "hidden_act": "gelu",
      "hidden_dropout_prob": 0.1,
      "hidden_size": 1024,
      "initializer_range": 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",
      "type_vocab_size": 2,
      "vocab_size": 21128
    }
    ```

- **preformance**
    - **chinese-macbert-large -> chinese-macbert-large** :
      0.75768 (Public Score)
    - **chinese-macbert-large -> chinese-roberta-wwm-ext-large** :
      0.77124 (Public Score)
    - **chinese-roberta-wwm-ext-large -> chinese-macbert-large** :
      0.77124 (Public Score)
    - **chinese-roberta-wwm-ext-large -> chinese-roberta-wwm-ext-large** :
      0.78752 (Public Score)

Q3: Curves

- **Loss function :**
  The output of QA is the logit of start position and logit of end position.
    1. First calculate crossentropy for start logits and end logits between predict and one hot target.
    2. Use sum of start crossentropy and end crossentropy devided by 2 as loss for single data.
    3. Take the average of single data's loss as loss of dataset.

epoch	Accuracy (%)	Loss (CrossEntropy)
0.05	70.08973081	1.510970831
0.1	74.0777667	1.408432245
0.16	72.61548687	1.554771781
0.21	75.37387836	1.328793049
0.26	75.07477567	1.258555651
0.31	77.666999	1.307369232
0.37	77.46759721	1.395266056
0.42	78.19873712	1.308178902
0.47	78.73047524	1.41177547
0.52	79.06281157	1.324703574
0.58	80.4253905	1.244641185
0.63	79.46161515	1.327824831
0.68	79.82718511	1.309601784
0.73	79.89365238	1.282012939
0.78	81.32269857	1.305575371
0.84	82.08707212	1.114360332
0.89	82.5855766	1.163561106
0.94	82.5855766	1.175486088
0.99	82.41940844	1.098547697



Q4: Pretrained vs Not Pretrained

In the pretrained, I load model as :

- mc

```
model = AutoModelForMultipleChoice.from_pretrained(
    model_args.model_name_or_path,
    from_tf=bool(".ckpt" in model_args.model_name_or_path),
    config=config,
    cache_dir=model_args.cache_dir,
    revision=model_args.model_revision,
    use_auth_token=True if model_args.use_auth_token else None,
)
```

- qa

```
model = AutoModelForQuestionAnswering.from_pretrained(
    model_args.model_name_or_path,
    from_tf=bool(".ckpt" in model_args.model_name_or_path),
    config=config,
    cache_dir=model_args.cache_dir,
    revision=model_args.model_revision,
    use_auth_token=True if model_args.use_auth_token else None,
)
```

In the not Pretrained, I load model as :

- mc

```
model = AutoModelForMultipleChoice.from_config( config )
```

- qa

```
model = AutoModelForQuestionAnswering.from_config( config )
```

- **nopretrained-mc**
    - **Loss function** : Cross Entrophy
    - **optimization algorithm** : AdamW
    - **learning rate** : 3e-5
    - **batch size**: 2 (**per_gpu_train_batch_size** : 1, **gradient_accumulation_steps** : 2)
    - **config** :

```
{
    "architectures": [
      "BertForMaskedLM"
    ],
    "attention_probs_dropout_prob": 0.1,
    "bos_token_id": 0,
    "directionality": "bidi",
    "eos_token_id": 2,
    "hidden_act": "gelu",
    "hidden_dropout_prob": 0.1,
    "hidden_size": 512,
    "initializer_range": 0.02,
    "intermediate_size": 2048,
    "layer_norm_eps": 1e-12,
    "max_position_embeddings": 512,
    "model_type": "bert",
    "num_attention_heads": 8,
    "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",
    "type_vocab_size": 2,
    "vocab_size": 21128
  }
```

- **nopretrained-qa**
  - **Loss function** : Cross Entropy
  - **optimization algorithm** : AdamW
  - **learning rate** : 3e-5
  - **batch size**: 2 (**per_gpu_train_batch_size** : 1, **gradient_accumulation_steps** : 2)
  - **config** :

```
{
    "architectures": [
      "BertForMaskedLM"
    ],
    "attention_probs_dropout_prob": 0.1,
    "bos_token_id": 0,
    "directionality": "bidi",
    "eos_token_id": 2,
    "hidden_act": "gelu",
    "hidden_dropout_prob": 0.1,
    "hidden_size": 512,
    "initializer_range": 0.02,
    "intermediate_size": 2048,
    "layer_norm_eps": 1e-12,
    "max_position_embeddings": 512,
    "model_type": "bert",
    "num_attention_heads": 8,
    "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",
    "type_vocab_size": 2,
    "vocab_size": 21128
  }
```

- **preformance**
  - **chinese-roberta-wwm-ext-large -> nopretrained-qa** :
    0.05605 (Public Score)
  - **nopretrained-mc -> chinese-roberta-wwm-ext-large** :
    0.44846 (Public Score)
  - **the best of ptrtrained** :
    0.78752 (Public Score)
  - **conclusion** :
    The score drops a lot with nopretrain model, especially for the complicated target like qa.

Q5: Bonus: HW1 with BERTs

- **Slot tagging**
  - **Loss function** : Cross Entropy
  - **optimization algorithm** : AdamW
  - **learning rate** : 3e-5
  - **batch size**: 2
    - **per_gpu_train_batch_size** : 1
    - **gradient_accumulation_steps** : 2
  - **config** :

```
{
  "_name_or_path": "./roberta_load/pytorch_model.bin",
  "architectures": [
    "BertForTokenClassification"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "classifier_dropout": null,
  "directionality": "bidi",
  "eos_token_id": 2,
  "finetuning_task": "ner",
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 1024,
  "id2label": {
    "0": "B-date",
    "1": "B-first_name",
    "2": "B-last_name",
    "3": "B-people",
    "4": "B-time",
    "5": "I-date",
    "6": "I-people",
    "7": "I-time",
    "8": "O"
  },
  "initializer_range": 0.02,
  "intermediate_size": 4096,
  "label2id": {
    "B-date": 0,
    "B-first_name": 1,
    "B-last_name": 2,
    "B-people": 3,
    "B-time": 4,
    "I-date": 5,
    "I-people": 6,
    "I-time": 7,
    "O": 8
  },
  "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
}
```

- **preformance** :
  - **bert** :
    public : 0.82412
    private : 0.83762
  - **rnn** :
    rnn public : 0.81983
    rnn private : 0.83011