[ADL F113]

HW1

Student id : r13922154 章程睿

Q1. Data Processing

■ Tokenizer:

The models use the tokenization algorithm from the "hfl/chinese-lert-base" model, which is likely based on a variant of WordPiece. This algorithm is specifically designed for Chinese text processing:

1. Character-level tokenization: Chinese characters are treated as basic units.
2. WordPiece segmentation: Common character combinations are recognized as single tokens.
3. Out-of-vocabulary handling: Rare words are split into more common subwords.
4. Special token addition: Special tokens like [CLS], [SEP] are added to mark the beginning, separation, and end of sentences.

Key features:

- Preserves the meaning of individual Chinese characters
- Captures common word structures in Chinese
- Balances vocabulary size and representation ability

■ Answer span:

1. I use an offset mapping to associate character positions with token positions. In the prepare_train_features function, we iterate through the offset mapping to find the tokens that contain the answer's start and end characters. The indices of these tokens become the new start

and end positions. This process takes into account padding and special tokens, ensuring we accurately locate the answer within the tokenized input.

2. After the model predicts probability distributions, I apply a series of rules in the postprocess_qa_predictions function to determine the final answer span:

- We first select the N start and end positions with the highest probabilities.
- We then consider all possible combinations of these positions, applying constraints (e.g., start position must be before or equal to end position, answer length must not exceed the maximum allowed length).
- For each valid span, we calculate its score (sum of log probabilities for start and end positions).
- The span with the highest score is chosen as the final answer.
- We also implement logic to handle unanswerable questions by comparing the score of the best non-null answer with a threshold to decide whether to return a null answer.

This approach allows us to accurately locate answers within the tokenized input while handling various edge cases, such as answers spanning multiple tokens or questions that may have no answer.

Q2. Model with Berts and their variants

■ Describe:

The first model I use is **BERT (bert-base-chinese) :**

	Paragraph selection	**Span selection**
Loss function	Cross entropy	Cross entropy
optimizer	AdamW	AdamW
Learning rate	3e-5	3e-5
Batch size per device	1	1

Gradient_accumulation_step	2	2
epochs	1	3
Max_seqence_length	512	512

Performance on validation data:

Paragraph selection accuracy	**95.51345962113659 %**
Span selection exact match score	**78.46460618145564 %**

Describe of the bert-base-chinese:

BERT-base-Chinese is a pre-trained language model based on the original BERT (Bidirectional Encoder Representations from Transformers) architecture. It was specifically trained on Chinese text data to understand and generate Chinese language.

1. Architecture: Uses the same architecture as BERT-base, with 12 layers, 768 hidden units, and 12 attention heads.
2. Vocabulary: Has a vocabulary of about 21,128 Chinese characters and word pieces.
3. Training data: Trained on large Chinese corpora, including Wikipedia and other web texts.
4. Input: Takes Chinese text as input, typically processed into word pieces.
5. Applications: Useful for various Chinese NLP tasks like text classification, named entity recognition, and question answering.
6. Fine-tuning: Can be fine-tuned on specific downstream tasks for improved performance.

■ Try another type of pre-trained LMs and describe:

The model I trained to get the highest score is **LERT(Chinese-lert-base)**

	Paragraph selection	**Span selection**
Loss function	Cross entropy	Cross entropy
optimizer	AdamW	AdamW

Learning rate	3e-5	3e-5
Batch size per device	1	1
Gradient_accumulation_step	2	2
epochs	1	3
Max_seqence_length	512	512

Performance on validation data:

Paragraph selection accuracy	**96.80957128614157%**
Span selection exact match score	**82.0538384845436%**

Describe of the Chinese-lert-base :

Chinese-LERT-base is a language model designed for Chinese natural language processing tasks. LERT stands for "Lexicon Enhanced Representation Transformer," indicating it incorporates lexicon information to enhance its understanding of Chinese text.

1. Architecture: Based on the BERT architecture, but with modifications to incorporate lexicon information.
2. Lexicon enhancement: Integrates Chinese lexicon knowledge, potentially improving its understanding of word meanings and relations.
3. Training: Trained on large-scale Chinese corpora, likely including web text and formal publications.
4. Applications: Suitable for various Chinese NLP tasks such as text classification, named entity recognition, and sentiment analysis.
5. Performance: May offer improvements over standard BERT models on certain Chinese language tasks, particularly those where lexical knowledge is beneficial.
6. Availability: Likely available through popular model repositories for use in research and applications.

key differences between LERT and BERT:

LERT (Lexicon Enhanced Representation Transformer) adds several

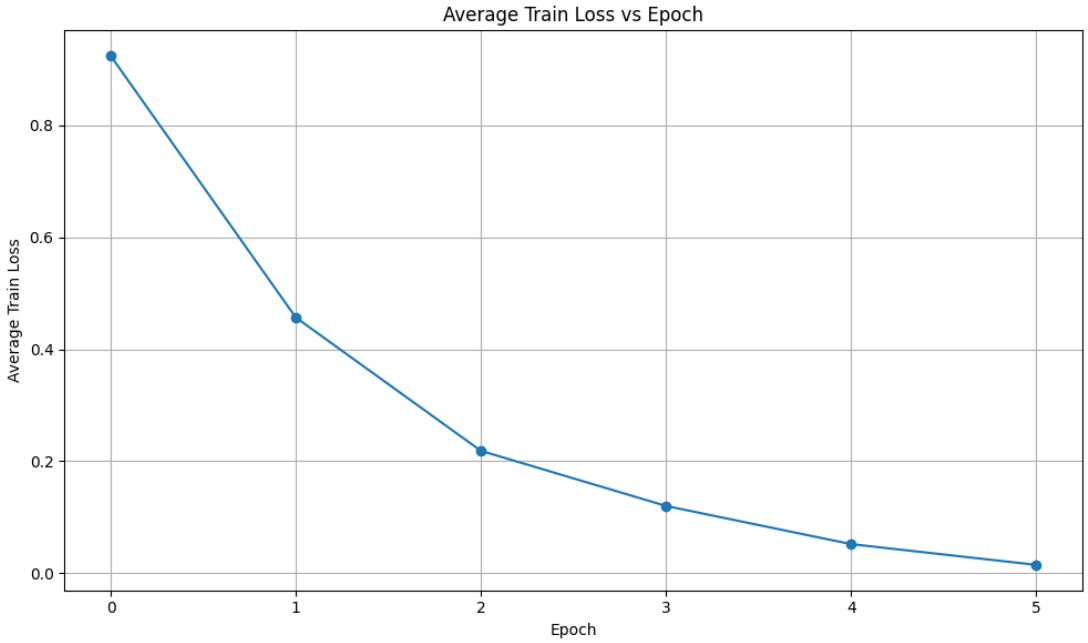
enhancements to the original BERT model:

1. Lexicon Enhancement: The core feature of LERT is the integration of lexical information. This means it doesn't rely solely on context to understand words, but also incorporates additional lexical knowledge.
2. Chinese Lexicon Knowledge: Specifically for Chinese, LERT incorporates knowledge about Chinese vocabulary. This may include word definitions, usage, synonyms, antonyms, and other lexical information.
3. Improved Word Meaning Understanding: By integrating lexical information, LERT may better understand the nuanced meanings and uses of words, especially when dealing with polysemous words or technical terms.
4. Grasp of Word Relationships: LERT is likely better at understanding semantic relationships between words, which can be advantageous in certain tasks.
5. Targeted Optimization: LERT's design may be more tailored to the characteristics of the Chinese language, potentially performing better when handling certain Chinese-specific linguistic phenomena.
6. Potential Performance Improvements: In some specific Chinese NLP tasks, especially those requiring deep lexical knowledge, LERT may outperform standard BERT models.

Q3. Curves:

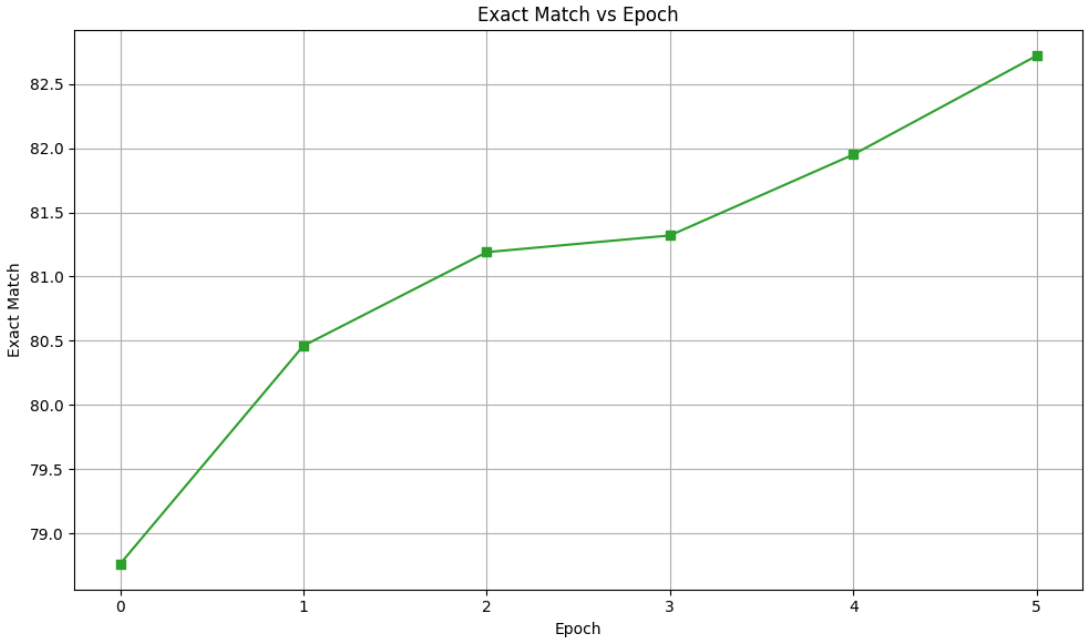
■ Learning curve of the loss value:

Below shows the loss curve on span selection “Validation Set”. The loss value is the average loss per example.



■ **Learning curve of the Exact Match metric value:**

Below shows the exact match metric learning curve on span selection “Validation Set”.



Q4. Pre-trained vs Not Pre-trained

I choose to train the paragraph selection model from scratch(without pretrained weights).

I use the recommend from claude AI to train the model.

Model Configuration:

- Utilizes a smaller BERT architecture with a hidden layer size of 256, 4 hidden layers, and 4 attention heads.
- This configuration is suitable for training from scratch while maintaining the basic structure of a transformer.

Training Configuration:

- Learning rate is set to 5e-4, which is higher than typically used when fine-tuning pre-trained models.
- The number of training epochs is increased to 10, as training from scratch usually requires more training time.
- Employs a linear learning rate scheduler with a warmup period.

Data Processing:

- The code retains the original data processing logic, including reading context information from JSON files.

Optimizer and Learning Rate Schedule:

- Uses the AdamW optimizer with a weight decay set to 0.01.
- Implements a linear learning rate scheduler that includes a warmup period.

Pre-trained vs Not pre-trained:

Model/Task performance (Validation)	paragraph selection(accuracy)
bert-base-chinese (pretrained)	**0.9513459**
Bert-base-uncased(scratch)	**0.4822200066467265**