Assignment 1

Please submit the code and report (in pdf format) on Blackboard system before 23:59 9th. Oct.. Report can be written either in English or Chinese. Submit your report `studentID_Name.pdf` for this assignment, and your code `studentID_Name.ipynb` for reproduction. If you need to attach other files, please submit it in `studentID_Name.zip`.

Q1 [30 points]

Select three topics from the NLP area, choose one benchmark dataset for each topic. Briefly introduce its input, output, evaluation method, and list the top-2 models/methods with the best performance on this benchmark dataset, and briefly introduce the model/method and its performance.

从NLP领域选择三个方向，每个方向选择一个benchmark dataset。简单介绍其输入、输出、评估方法，并列出在该benchmark dataset上表现最佳的前两个模型/方法，简单介绍该模型/方法以及其表现。

Reference (including but not limited to):

> Browse the State-of-the-Art in Machine Learning | Papers With Code (Recommended!!)
>
> Open LLM Leaderboard 2 - a Hugging Face Space by open-llm-leaderboard
>
> MTEB Leaderboard - a Hugging Face Space by mteb

Q2 [70 points]

1. Learn about Tokenizer [20 points]

Currently, the mainstream tokenizer training algorithms in use include BPE (Byte-Pair Encoding), BBPE (Byte-level BPE), Unigram, WordPiece, SentencePiece. Among them, BBPE is famous for being used by the GPT-2 tokenizer, WordPiece is used by the BERT tokenizer, and SentencePiece is the algorithm used by the Llama tokenizer.

Please search and learn about the above algorithms on your own, and then introduce them in detail from their principles, calculation methods, optimization objectives, and other aspects.

> Notice 1: You don't need to list all the details of these algorithms, just provide an introduction. Our grading criteria focus on whether the introduction is comprehensive and accurate.
>
> Notice 2: You may be confused about the relationships between these algorithms. The relationships between these algorithms are not entirely parallel; they may be interconnected. For instance, the principles of WordPiece and BPE are quite similar, and one of the training algorithms for SentencePiece is the BPE algorithm.

请你自行搜索并学习以上算法（BPE、BBPE、Unigram、WordPiece、SentencePiece），然后从它们的原理、计算方法、优化目标等方面展开介绍。

> 注意一：你不需要列举出所有关于算法的细节，我们的评分标准在于你的介绍是否全面且准确
>
> 注意二：你可能会对这些算法之间的关系感到困惑。他们之间的关系并不是完全平行的，例如WordPiece与BPE的原理十分相似，而SentencePiece采用BPE和Unigram作为训练算法

2. Train a Tokenizer [20 points]

Using the provided corpus `data.txt`, freely choose any algorithm, training method, and hyper-parameters to train a tokenizer with a vocabulary size of 6000. It is recommended to use tools such as the `huggingface tokenizer`, `NLTK`, or `sentencepiece`， you are also welcomed to implement the algorithm by numpy.

使用提供的语料`data.txt`，自由选择以上任意一种算法或训练方式，以及其超参数，训练一个词表大小为6000的分词器。推荐使用`huggingface tokenizer`、`NLTK`、`sentencepiece`等工具包，你也可以使用numpy手动实现。

Reference (including but not limited to):

> Tokenizers (huggingface.co)
>
> NLTK :: Natural Language Toolkit
>
> sentencepiece/python at master · google/sentencepiece (github.com)

3. Train a Word Embedding Model [20 points]

On the tokenizer you trained, use the CBOW or skip-gram algorithm to train a word embedding model, and then use the t-SNE algorithm (using the `scikit-learn` toolkit and the `matplotlib` toolkit) for visualization (take only 100 random words). It is recommended to use toolkits such as `FastText` or `Gensim`.

在你得到的分词器上，使用CBOW或者skip-gram算法，训练一个word embedding模型，然后使用t-SNE算法（可使用`scikit-learn`工具包和`matplotlib`工具包）进行可视化（仅取随机100个词即可）。推荐使用`FastText`、`Gensim`等工具包。

Reference (including but not limited to):

> Word representations · fastText
>
> Word2Vec Model — gensim (radimrehurek.com)

4. More to Explore[10 points]

1. BytePiece

BytePiece is a tokenizer training algorithm with higher compression rate. Train a tokenizer using BytePiece, then use it to train a word embedding model and visualize the results using t-SNE. Compare the results with the tokenizer you used previously.

BytePiece是一个压缩率更高的分词器训练算法。使用BytePiece训练一个tokenizer，然后用它训练一个word embedding模型，使用t-SNE进行可视化。把结果与你前面使用的分词器进行对比。

Reference:

> https://kexue.fm/archives/9752
>
> https://github.com/bojone/bytepiece

2. Hand-written Implementation

Implement your tokenizer and word embedding using numpy and the standard library, and compare the results with the toolkit you previously used. (If you previously used handwritten code, compare it with the results of any open-source toolkit.)

使用numpy以及标准库实现你的tokenizer和word embedding，与你先前使用的工具包的效果进行对比。（如果你前面使用手写实现的代码，则与任意一个开源工具包的效果进行对比）