声明：我已知悉学校对于考试纪律的严肃规定，将秉持诚实守信宗旨，严守考试纪律，不作弊，不剽窃；若有违反学校考试纪律的行为，自愿接受学校严肃处理。

2018-2019 学年第二学期 COMP130137.01 《模式识别与机器学习》课程项目 2019 语言与智能技术竞赛-机器阅读理解

学号：xxxxxx , 姓名：xxxxxx，贡献: xx%，签名：

Abstract

Abstract

1 Overview

1.1 Introduction

Machine Reading Comprehension (MRC) means that let machine read context and then answer the question which is based on the material. MRC is an important frontier in the field of natural language processing and artificial intelligence. The competition focus on the problem which are hard to answer in 2018 Machine Reading Comprehension Technology Competition.

1.2 Assignment

For a given question q and its corresponding text form candidate document set $D = d1, d2, ..., dn$, the participation reading comprehension system is required to automatically analyze the problem and the candidate document, and output a text answer a that satisfies the question. The goal is that a can answer the question q correctly, completely, and concisely.

1.3 About Dataset

The dataset contains about 280,000 real problem from Baidu Research, and each of them correspond to 5 candidate document and manual answer. The data set is divided into a training set of 270,000 questions, a development set of approximately 3,000 question, and a test set of approximately 7,000 questions.

2 BERT and BiDAF Introduction

2.1 BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a new language representation model. BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, without substantial task-specific architecture modifications.Devlin et al. [2018]

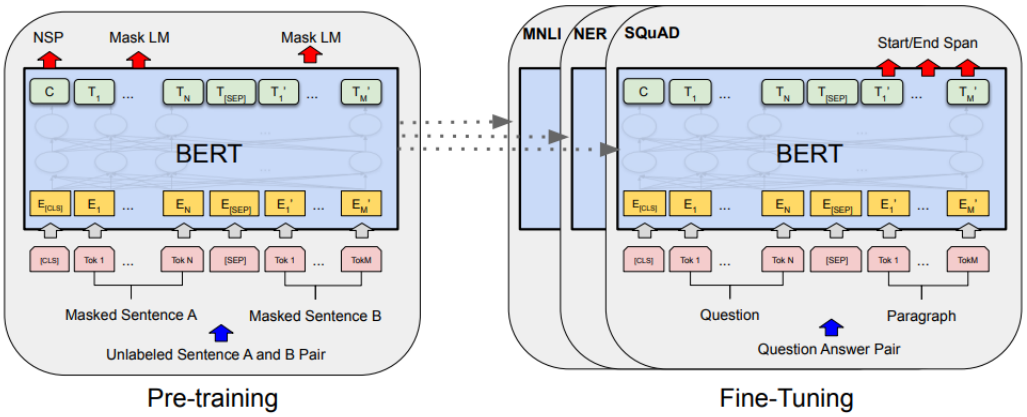

Figure 1: Overall pre-training and fine-tuning procedures for BERT

2.2 BiDAF

Bi-Directional Attention Flow (BiDAF) network is a hierarchical multi-stage architecture for modeling the representations of the context paragraph as different levels of granularity. BIDAF includes character-level, word-level, Seo et al. [2016]

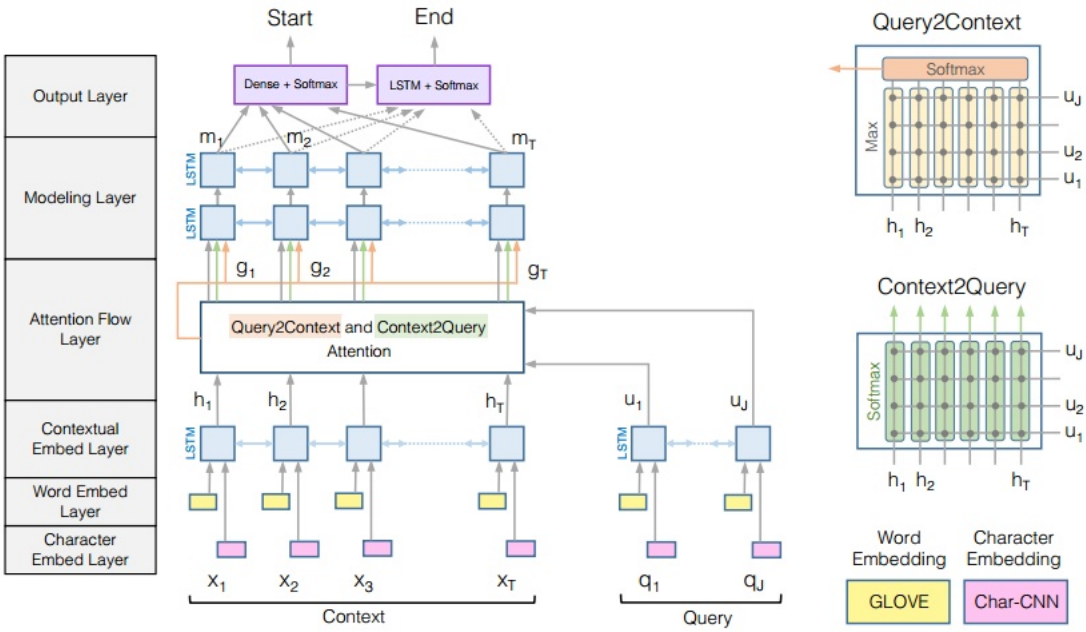

Figure 2: BiDirectional Attention Flow Model

3 Data Preprocessing

3.1 Label choice

The dataset is organized in .json format, which provides various labels about the query and context. After we observe some of them, however, we found that not all the labels is useful. For example, the label "answer" and "fake_answer" both provide corresponding answer about the query. But the former is generated manually, and the latter is extract from the context. For simplicity, we regard the latter as our target.

3.2 Text preprocessing

Considering the dataset contains real problem from Baidu Research, most of the context and query is Chinese. Besides, we found that the context contains lots of incorrelative characters like "20170113", "...", "jkefsgasscd". They don't provide any useful information about how to solve the problem. So we decide to omit all English letter and other non-Chinese character. We gain the text which only has Chinese character and punctuation.In my opinion, it can express the meaning of query and answer better, and it can improve the performance of our model.

Besides, it would take lots of time for BERT to maps query and context to embedding vector when the vocabulary is large. Some words like " 是"、" 你"、" 做" appears frequently, and most of the query and context can be expressded in a simply way. That is, we can use small dictionary to express the paragraph. In this assignment, we set 6000 as the size of the dictionary. It can be proved that BERT can generate corresponding vector in a short time.

3.3 Data format conversion

We have to convert data from text to vector that fit the input of the model. Each words in dictionary will embed in a 768-dimension vector.

4 Model

Our model is a multi-stage process and consists of two parts(Figure 3):

1. **BERT** maps each word in query and context to a vector
2. **BiDAF** maps query and context vector in the beginning and ending position of the answer.

4.1 BERT

4.2 BiDAF

Notations:

lenCon: the length of the context.

lenQue: the length of the query.

D_H: the size of the hidden layer.

BiDAF is the main part of this model. BiDAF is short for Bi-directional Attention Flow, which means it uses backward and forward information to achieve this task. In fact, the BiDAF model is a combination of Bi-LSTM and attention, which are both suitable for QA problem.

In this task, we reproduce BiDAF and makes some changes to match QA in Chinese better. Before we dive into BiDAF, we will first have a overview of this model. BiDAF is a model with six layers originally, but we only have four layers indeed. The original six layers are character embeddign layer, word embedding layer, contextual embedding layer, attention flow layer modeling layer and output layer, which in our model, the character embedding layer and word embedding layer are replaced by a pre-trained matrix provided by BERT.

Word embedding layer and character embedding layer: These two layers transform words and characters into embedding vectors. Since we have used BERT to do the embedding work, the importance of these layers decreases. Therefore, the real inputs are embedded matrixs produced by pretrained BERT. And we don't have enough resources to train BERT or even finetune it, we will simply fixed the output of BERT to do the embedding.

Contextual embedding layer: This layer is simply a BiLSTM which makes the model have a general view of the whole passage. Since BiLSTM is wildly used in many models, we will not introduce the structure of the model in details.

Attention flow layer: This layer is the most important layer in this model. Th ebasic idea of this layer is to get two vectors: c2q and q2c, which makes the model be aware of the connections between the

context and query. This layer is also the attention layer as mentioned in the name of BiDAF. This layer trys to make the context to be aware of the query and also to make the query be aware of the context. The computations in this layer is quite complicated.

To begin with, we have to mention the similarity matrix S(lenCon*lenQue). S[i][j] means the similarity between the i-th character in context and j-th character in query. And S[i][j]=W[C[i];Q[j];C[i]*Q[j]]+b.

After we have got the similarity matrix, we need to compute the q2c and c2q matrix. We do the computation in the following way:

$$a_t = softmax(S_{t:})U_{:t} = \sum_j a_{tj}U_{:j}$$

. To be explicit, this matrix computes the sum of the similarity of each word in the context and the query. And finally, we get a new matrix of (2*D_H , lenCon). In this way, we get the representations of the c2q. As for q2c, the idea is quite the same as c2q. The only difference is the direction to use this matrix.

$$b = softmax(max(S))h = \sum_t b_t H_{:t}$$

And we get another matrix: (s*D_H* lenCon). Some difference here is that the length of context is much longer than the length of the query, so we need the most connnected word between context and query. That's where the max function comes from. Modeling layer: Like Contextual layer, this layer is a combination of two BiLSTM. It servers for the comprehensions of the information fetched in the formor layers. Output layer: Work for this layer is simple: using the data to make predictions of the starts and ends of the answers.

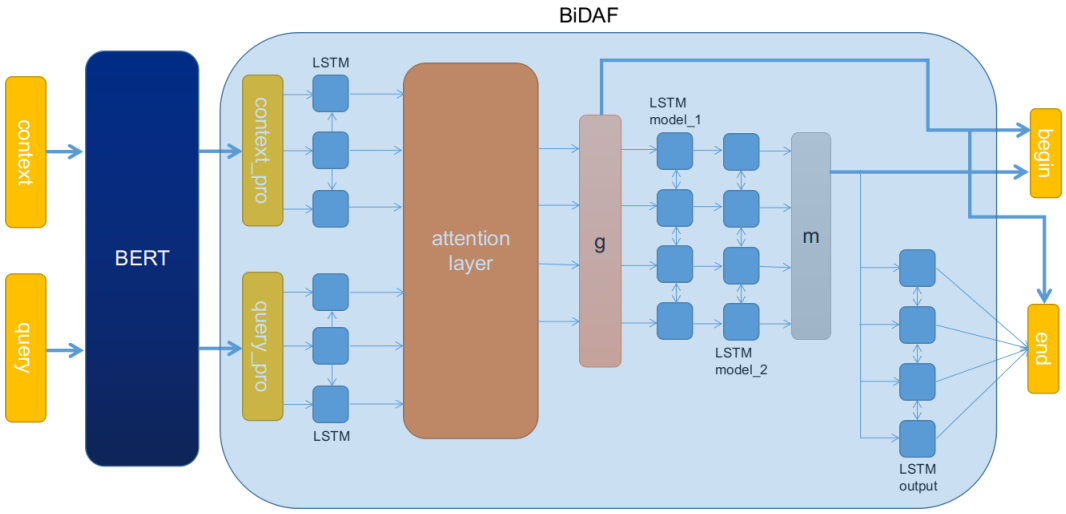

Figure 3: Model combined by BERT and BiDAF

5

选择下面任一项目进行实现。

5.1 RNN 速度改进

改进 LSTM 或 GRU 模型，提高模型并行化能力。

5.1.1 参考文献

1. Quasi-Recurrent Neural Networks
https://openreview.net/forum?id=H1zJ-v5xl

2. Simple Recurrent Units for Highly Parallelizable Recurrence
   https://arxiv.org/abs/1709.02755
3. Phased LSTM: Accelerating Recurrent Network
4. Skip RNN: Learning to Skip State Updates in Recurrent Neural Networks
   https://openreview.net/forum?id=HkwVAXyCW
5. Yu et al., Learning to Skim Text, ACL 2017
6. Neural Speed Reading via Skim-RNN
   https://openreview.net/forum?id=Sy-dQG-Rb
7. Variable Computation in Recurrent Neural Networks
   https://arxiv.org/abs/1611.06188

5.2 Few-shot Learning for Text Classification

Consider a supervised learning task T , FSL deals with a data set $D = \{D_{train}, D_{test}\}$ consisting of training set $D_{train} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{I}$ where I is small and test set $D_{test} = \{x_{test}\}$. Usually, people consider the N-way-K-shot classification task where D_{train} contains $I = KN$ examples from N classes each with K examples.

Dataset: https://github.com/Gorov/DiverseFewShot_Amazon

5.2.1 参考文献

1. Learning to Compare: Relation Network for Few-Shot Learning
   https://arxiv.org/pdf/1711.06025v2.pdf
2. A CLOSER LOOK AT FEW-SHOT CLASSIFICATION
   https://openreview.net/pdf?id=HkxLXnAcFQ
3. Advances in few-shot learning: a guided tour
   https://towardsdatascience.com/advances-in-few-shot-learning-a-guided-tour-36bc10a68b77
4. Generalizing from a Few Examples: A Survey on Few-Shot Learning
   https://arxiv.org/pdf/1904.05046.pdf
5. Advances in few-shot learning: reproducing results in PyTorch
   https://towardsdatascience.com/advances-in-few-shot-learning-reproducing-results-in-pytorch-aba70dee541
6. Few-Shot Text Classification with Induction Network
   https://arxiv.org/pdf/1902.10482.pdf

6 实现要求

项目实现需基于开源项目 fastNLP（https://github.com/fastnlp/fastNLP）进行。如果目前的 fastNLP 功能不足以实现某个算法，可以随便修改。之后也欢迎为 fastNLP 贡献 PR。具体要求如下：

1. 程序正确性，可顺利运行；
2. 加分项：为 fastNLP 提 PR，并通过单元测试。
   (a) GIT 操作及 PR 操作：https://github.com/fastnlp/fastNLP/wiki/怎样使用Git进行开发
   (b) 代码规范参考：https://github.com/fastnlp/fastNLP/wiki/fastNLP-代码规范
   (c) 单元测试说明：https://github.com/fastnlp/fastNLP/wiki/fastNLP测试说明

7 项目报告

项目报告作为判断项目质量和工作量的主要依据，请务必详细在报告中描述项目的主要亮点。中英文均可，不少于 5 页。报告包含以下内容：

1. 问题描述、动机

2. 方法和技术
3. 实验设计
4. 结果分析
5. 相关工作对比、分析

8 报告的格式信息

项目报告采用 NeurIPS 会议论文格式，具体信息如下：

The style files for NeurIPS and other conference information are available on the World Wide Web at

http://www.neurips.cc/

The file `neurips_2018.pdf` contains these instructions and illustrates the various formatting requirements your NeurIPS paper must satisfy.

The formatting instructions contained in these style files are summarized in Sections 9, 10, and 11 below.

9 General formatting instructions

The text must be confined within a rectangle 5.5 inches (33 picas) wide and 9 inches (54 picas) long. The left margin is 1.5 inch (9 picas). Use 10 point type with a vertical spacing (leading) of 11 points. Times New Roman is the preferred typeface throughout, and will be selected for you by default. Paragraphs are separated by ½ line space (5.5 points), with no indentation.

Please pay special attention to the instructions in Section 11 regarding figures, tables, acknowledgments, and references.

10 Headings: first level

All headings should be lower case (except for first word and proper nouns), flush left, and bold.

First-level headings should be in 12-point type.

10.1 Headings: second level

Second-level headings should be in 10-point type.

10.1.1 Headings: third level

Third-level headings should be in 10-point type.

Paragraphs There is also a `\paragraph` command available, which sets the heading in bold, flush left, and inline with the text, with the heading followed by 1 em of space.

11 Citations, figures, tables, references

These instructions apply to everyone.

11.1 Citations within the text

The `natbib` package will be loaded for you by default. Citations may be author/year or numeric, as long as you maintain internal consistency. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

The documentation for `natbib` may be found at

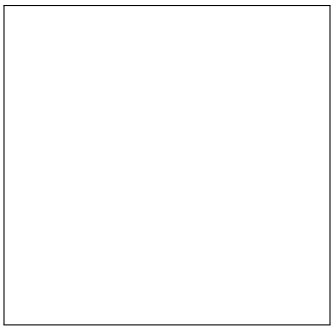

Figure 4: Sample figure caption.

http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{adams1995hitchhiker} investigated\dots
```

produces

Collobert and Weston [2008] investigated...

If you wish to load the `natbib` package with options, you may add the following before loading the `neurips_2018` package:

```
\PassOptionsToPackage{options}{natbib}
```

If `natbib` clashes with another package you load, you can add the optional argument `nonatbib` when loading the style file:

```
\usepackage[nonatbib]{neurips_2018}
```

11.2 Footnotes

Footnotes should be used sparingly. If you do require a footnote, indicate footnotes with a number[1] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).

Note that footnotes are properly typeset *after* punctuation marks.[2]

11.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction. The figure number and caption always appear after the figure. Place one line space before the figure caption and one line space after the figure. The figure caption should be lower case (except for first word and proper nouns); figures are numbered consecutively.

You may use color figures. However, it is best for the figure captions and the paper body to be legible if the paper is printed in either black/white or in color.

11.4 Tables

All tables must be centered, neat, clean and legible. The table number and title always appear before the table. See Table 1.

[1] Sample of the first footnote.

[2] As in this example.

Table 1: Sample table title

	Part	
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

Note that publication-quality tables *do not contain vertical rules.* We strongly suggest the use of the `booktabs` package, which allows for typesetting high-quality, professional tables:

https://www.ctan.org/pkg/booktabs

This package was used to typeset Table 1.

12 Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

References

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, pages 160–167, 2008. doi: 10.1145/1390156.1390177. URL http://doi.acm.org/10.1145/1390156.1390177.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016. URL http://arxiv.org/abs/1611.01603.