文档类别

杭州海康威视数字技术股份有限公司

文档编号

算法预研报告

|  |  |
| --- | --- |
| 编制 |  |
| 审批 |  |

密级级别：[内部公开]

生效时间： 2021年5月25日

保密期：无

杭州海康威视数字技术股份有限公司 版权所有

**目录**

[1. 简介 3](#_Toc75945732)

[1.1 编写目的 3](#_Toc75945733)

[1.2 背景 3](#_Toc75945734)

[1.3 预期成果 3](#_Toc75945735)

[2. 技术介绍 3](#_Toc75945736)

[2.1 基本术语及定义 3](#_Toc75945737)

[2.2 算法定义和术语 3](#_Toc75945738)

[2.3 技术综述 3](#_Toc75945739)

[3. 预研过程 4](#_Toc75945740)

[3.1 算法方案 4](#_Toc75945741)

[3.2 训练方式 4](#_Toc75945742)

[3.3 方案论证 4](#_Toc75945743)

[4. 实验结果及分析 4](#_Toc75945744)

[5. 总结 4](#_Toc75945745)

[6. 参考文献： 5](#_Toc75945746)

[7. 修订记录 5](#_Toc75945747)

# 简介

## 编写目的

本文档旨在调研对于长文本表征的实现方式，为下游的长文本任务提供前提。

## 背景

在之前的NLP任务中，bert对于文本的表征有长度512的限制，并且由于bert的位置编码是学习得来的，它不支持对512位以后的位置编码的表达。对于长文本来说，512位以后的文本被忽略了。我们需要对位置编码做出改变使得他们能够支持对长文本的表征。

## 预期成果

本次以预研将介绍两种方式来改变bert原有的位置编码方式，使得bert模型能够embedding长文本，但由于显卡限制，maxlen不能设置太大，所以现在maxlen限制为1024。

# 技术介绍

## 基本术语及定义

* maxlen：每个bert模型会设置一个maxlen数值，它使得模型只能embedding前maxlen个token。
* embedding：通过模型将token转化为向量。
* token：指文本在词库当中的索引。
* 位置编码：bert模型不仅会对词做embedding，也会对每个位置做embedding，而每个位置的embedding便为其位置编码

## 算法一

本节将复现位置编码算法。并概述技术综述，算法方案，训练方式。

### 技术综述

初始模型仅为一个基础的roberta中文模型，在build\_transformer\_model()函数中加上hierarchical\_position=True 即可使得模型突破自身maxlen的限制。可以先使用预训练任务对模型进行训练，再进行下游任务的fintune（这里由于显卡限制，原模型的maxlen和下游任务的maxlen皆为1024，所以省去了预训练任务）。训练好的模型仅仅是一个能够生产不止1024位位置编码的模型，针对不同下游任务还需做相应的调整。该模型的input就是一段文本，通过已训练好的模型转换成embedding，然后进行下游任务。

### 算法方案

层次分解(finetune.py)是在position embedding层做了层次分解，因为显存限制，maxlen为1024，原模型roberta的maxlen也为1024，所以这里没有继续pre-train，但如果maxlen继续变大的话，先pre-train或许会有更好的效果。

在命令行输入 python fintune.py 或者sh finetune.sh进行训练。

### 基本参数

* maxlen：由于显存限制，一般设置为1024，如果在大显存卡上训练，可以加大maxlen数值。maxlen直接关乎model截取多长的token。
* batch\_size: 默认为8。
* epoch:默认为50，可根据是否收敛提前结束。
* hierarchical\_position：选择为True。

### 训练方式

在trainset(qianyu/shengtao/roformer-main/CAIL2019-SCM/train.json)上训练，在testset(qianyu/shengtao/roformer-main/CAIL2019-SCM/test.json)上验证，通过一个Triplet Network，得到每个文本的embedding，对每个样本的正负样例对比相似度，评估结果为accuracy。

### 数据集

本次数据集来源于CAIL-2019，每个样本有三个文本，分别构成一对正样例和一对负样例。例如有文本A、B、C，AB为一对正样例，AC为一对负样例。

### 实验结果及分析

最终结果为 final test acc: 0.668620。baseline为0.64。较baseline有所提升，但是由于只改变位置编码，提升并不明显。

## 算法二

本节将复现位置编码算法。并概述技术综述，算法方案，训练方式。

### 技术综述

初始模型为roformer预训练中文模，可以直接进行下游任务的fintune（这里由于显卡限制，原模型的maxlen和下游任务的maxlen皆为1024）。训练好的模型仅仅是一个能够生产不止1024位位置编码的模型，针对不同下游任务还需做相应的调整。该模型的input就是一段文本，通过已训练好的模型转换成embedding，然后进行下游任务。

### 算法方案

roformer(finetune\_scm.py)是在position embedding层把原本的绝对位置编码方式改为了RoPE编码，因为显存限制，maxlen为1024，原模型roformer的maxlen也为1024，所以这里没有继续pre-train，但如果maxlen继续变大的话，先pre-train或许会有更好的效果。

在roformer-main目录下，命令行输入 python fintune\_scm.py进行训练。

### 基本参数

* maxlen：由于显存限制，一般设置为1024，如果在大显存卡上训练，可以加大maxlen数值。maxlen直接关乎model截取多长的token。
* batch\_size: 默认为8。
* epoch:默认为50，可根据是否收敛提前结束。

### 训练方式

在trainset(qianyu/shengtao/roformer-main/CAIL2019-SCM/train.json)上训练，在testset(qianyu/shengtao/roformer-main/CAIL2019-SCM/test.json)上验证，通过一个Triplet Network，得到每个文本的embedding，对每个样本的正负样例对比相似度，评估结果为accuracy。

### 数据集

本次数据集来源于CAIL-2019，每个样本有三个文本，分别构成一对正样例和一对负样例。例如有文本A、B、C，AB为一对正样例，AC为一对负样例。

### 实验结果及分析

最终结果为 final test acc: 0.668620。baseline为0.64。较baseline有所提升，但是由于只改变位置编码，提升并不明显。

# 总结

由于此算法仅仅改进了位置编码，所以提升效果并不是特别明显，但是已达到想要对长文本做embedding的预期。如果预训练模型能够对专有领域语料库进行pretrain，效果更佳。并且用较新的sbert或者simCSE范式对模型进行fintune，或许能够更好的提升。当然其他的trick比如文本增强之类的也能有所提升。

# 参考文献：

[1] 苏剑林. (Mar. 23, 2021). 《Transformer升级之路：2、博采众长的旋转式位置编码 》[Blog post]. Retrieved from https://kexue.fm/archives/8265

[2] 苏剑林. (Dec. 04, 2020). 《层次分解位置编码，让BERT可以处理超长文本 》[Blog post]. Retrieved from https://kexue.fm/archives/7947

# 修订记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **变更时间** | **版本** | **变更人** | **审批人** | **变更说明** |
| 1 | 2022.5.19 | V 0.1.0 | 边圣陶 |  | 新建 |