# 基于预训练语言模型的繁体古文自动句读研究

**唐雪梅[1] 苏祺[2*] 王军[1] 陈雨航[1] 杨浩[3]**

1北京大学，信息管理系，北京市；
2北京大学，教育部计算语言学重点实验室，北京市；
3北京大学，儒藏编纂与研究中心，北京市
{sukia,junwang,yanghao2008}@pku.edu.cn{tangxuemei,1700016603}@stu.pku.edu.cn

## 摘要

未经整理的古代典籍不含任何标点，不符合当代人的阅读习惯，古籍断句标点之后有助于阅读、研究和出版。本文提出了一种基于预训练语言模型的繁体古文自动句读框架。本文整理了约10亿字的繁体古文语料，对于训练语言模型进行增量训练，在此基础上上实现古文自动句读和标点。实验表明经过大规模繁体古文语料增量训练后的语言模型具备更好的古文语义表示能力，能够有助提升繁体古文自动句读和自动标点的效果。融合了增量训练模型之后，古文断句F1值达到95.03%，古文标点F1值达到了80.18%，分别比使用未增量训练的语言模型提升1.83%和2.21%。为解决现有篇章级句读方案效率低的问题，本文改进了前人的串行滑动窗口方式，在一定程度上提高了句读效率，并提出一种新的并行滑动窗口方式，能够高效准确地进行长文本自动句读。

**关键词：** 自动句读 ； 自动标点 ； 预训练语言模型

# Automatic Traditional Ancient Chinese Texts Segmentation and Punctuation Based on Pre-training Language Model

**Xuemei Tang[1] Qi Su[2] Jun Wang[1] Yuhang Chen[1] Hao Yang[3]**

1Department of Information management , Peking University;
2MOE Key Lab of Computational Linguistics, School of EECS , Peking University
3Editorial and Research Center of Confucian Canon , Peking University
{sukia,junwang,yanghao2008}@pku.edu.cn{tangxuemei,1700016603}@stu.pku.edu.cn

## Abstract

Unrecognized ancient books do not contain any punctuation, which doesn't meet reading habits of modern people. Punctuation in ancient books is helpful for reading, research and publication. We propose an automatic punctuation framework of traditional ancient Chinese text based on pre-training language model. In this paper, we build a traditional Chinese ancient corpus, contain about 1 billion characters. Based on the corpus, we incremental training a language model. Experiment results show that the incremental training language model has better semantic representation ability for ancient Chinese, and can help to improve the performance of automatic text segmentation and punctuation of traditional ancient Chinese text . After integrating the incremental training model, the F1 of text segmentation reaches 95.03%, and the F1 of text punctuation reaches 80.18%, which is 1.83% and 2.21% higher than the language model without incremental training. In order to solve the problem of low efficiency of the existing text level segmentation, we improve the previous serial sliding window mode, which raises the efficiency of text segmentation to a certain extent, and

第二十届中国计算语言学大会论文集，第678页-第688页，呼和浩特，中国，2021年8月13日至15日。

we propose a new parallel sliding window mode, which can efficiently and accurately segment long text automatically.

**Keywords:** Automatic texts segmentation , Automatic Punctuation , Pre-training Language Model

## 1 引言

中华文明历史悠久，古典文籍浩如烟海。古籍具有极高的文献价值和学术价值，古籍整理不仅是连接现代和历史的桥梁，而且有利于民族文化的传承和研究。而古人在著书时一般不使用标点，现存的许多古籍也没有断句和标点，这给读者阅读学习和学者研究古籍造成了障碍。所谓“凡训蒙，须讲究，详训诂，明句读”，即是说句读是古人求学问道的基础。传统的古籍句读工作主要依靠人工，但人工句读对标注者的古汉语素养要求较高，一般人难以胜任。且中国古代典籍数量众多，人工句读效率低，短时间内无法完成批量典籍的句读工作。计算机自动句读可以有效地解决以上两个问题。古文自动句读是指根据古代汉语句子特点，结合现代汉语的标点符号用法，让计算机自动切割、断开连续的文本字符序列为句，然后加标点的过程 (黄水清and 王东波, 2017)。

古文自动句读经历30多年的发展，从基于规则的方法逐渐发展到基于深度学习的方法，但因目前没有公开的大规模的繁体古文语料库或者数据集，且整理过的古籍散落在不同的语料库或者出版社数据库，难以收集到大量整理过的繁体古籍文本，所以目前古文自动断句的研究基本都是针对简体汉字文本，如 (王博立et al., 2017; 胡韧奋et al., 2019; 俞敬松et al., 2019)等人的研究。而现存很多未被整理的古籍都是繁体汉字，若将繁体转为简体再做句读，繁简转化的错误可能会延续到句读的结果中。同时现在常用在古籍任务中的预训练语言模型(Devlin et al., 2018; Liu et al., 2019)都有固定的词表，词表中包含的繁体字较少，在词表之外繁体字会被替换成特殊字符造成语义的缺失，从而影响任务效果。所以构建一个专门用于繁体古文的句读模型是有必要的。断句之后的古籍文本方便阅读研究，标点之后的文本有助于整理出版，现有研究较多集中在自动断句 (程宁et al., 2020; 胡韧奋et al., 2019)，俞敬松等 (2019)虽然同时关注自动断句和自动标点，但用于自动标点的训练语料规模很小，标点效果并不理想；释贤超等(2018)在不同朝代的不同类型语料上进行自动标点研究，但其模型泛化能力有限。另一方面未经整理的古籍文本篇幅较长，整篇文章连成整体居多，篇章级句读是应用环境下必须解决的问题。现有的研究较少涉及篇章级的断句，胡韧奋等 (2019)的断句模型以段落为单位，俞敬松等 (2019)提出了一种串行滑动窗口方式处理长文本句读，但是该方法的句读效率比较低。

本文的主要工作有以下三项：

1) 本文整理了约10亿字的繁体古文语料，基于整理的语料增量训练BERT(Bidirectional Encoder Representation from Transformers) (Devlin et al., 2018)模型得到繁体古文预训练语言模型；

2)基于繁体古文预训练模型，利用高质量带标点繁体古文语料微调预训练语言模型，实现繁体古文的自动句读和自动标点。

3) 在前人工作基础之上，本文改进数据串行滑动窗口方式处理篇章级句读，在一定程上提高了运行效率；同时本文提出了一种数据并行的滑动窗口方案，不仅保证了自动句读的准确率，而且大幅度提高了篇章级句读的运行速率。

## 2 相关研究

古文自动句读的研究大致经历了三个发展阶段：分别是基于规则的阶段、基于统计方法的阶段以及基于深度学习的阶段。

黄建年等(2008)总结农业古籍的断句标点规则，包括句法特征、词法特征、引文特征等，利用规则在农业古籍上进行测试，断句的准确率为48%。基于规则的方法简单易于理解，但是需要专家建立规则库，不仅费时费力，且规则的覆盖面有限，只能用于处理小规模文本。


本研究得到国家自然科学基金国际重点合作项目“中国儒家学术史知识图谱构建研究”的支持(项目号：72010107003)

陈天莹等(2007)采用基于上下文的N-gram模型对古汉语文本做句子切分，在《论语》上达到了81%的召回率，52%的精确率。后续逐渐有学者将序列标注技术应用到自动断句任务中，黄瀚萱(2008)比较了基于字的条件随机场（CRF）和隐马尔科夫（HMM）模型在《孟子》《论语》上的断句效果，发现CRF模型优于HMM。张开旭等(2009)在CRF的基础上引入互信息和t-测试差，在《论语》和《史记》上训练断句，分别取得了0.762和0.682的F1值。张合等(**?**)基于六字位标集，采用层叠CRF对《老子》、《水经注》、《战国策》、《左传》、《赤壁赋》、《出师表》等进行断句和标点，低层CRF模型用于识别句子边界，高层CRF模型用于自动标点。基于统计的方法主要依靠人工特征模版，但是古籍文体风格多样，年代跨度大，很难构建一个适用于所有古籍的断句模版，导致统计模型的泛化能力较弱。

随着深度学习在自然语言处理领域的应用，陆续有学者将深度学习方法用于自动句读任务。循环神经网络是时序性结构，相比于卷积神经网络能够更好地处理长文本，经常用于序列标注任务。王博立(2017)在2.37亿字规模的训练集上训练双向GRU（门控循环神经网络）模型，该模型在古文上的断句F1值达75%。释贤超等(2018)在南北、隋、唐、宋、辽和明六个朝代的佛、道、儒典籍上比较了长短时记忆网络（LSTM）和卷积神经网络（CNN）的标点效果，实验表明LSTM的标点效果好于CNN，在唐代的语料上标点可以达到94.3%的精确率。古文分词需要建立在断句的基础之上，分步进行容易造成错误多级扩散，程宁等(2020)设计了断句、分词及词性一体化标注方法，利用Bi-LSTM模型同时训练断句、分词和词性标注三项任务，发现一体化标注方法对三个任务的F1值均有提升。

2018年谷歌提出了预训练语言模型BERT，通过精调在11项自然语言处理任务上的效果超过了之前的模型，自此古文句读模型也逐渐转向使用预训练语言模型阶段。俞敬松等(2019)利用3亿7千万殆知阁古文语料对BERT语言模型做断句和标点训练，分别在单一类别文本和复合文本上进行测试断句，达到了89.97%和91.67%的F1值，在单一文本上测试标点F1值达到了70.4%。胡韧奋等(2019)基于33亿字古汉语语料训练了古文BERT模型，并比较了BERT+FCL、BERT+CRF、BERT+CNN等序列标注方法在古文断句任务上的表现，发现BERT+CNN模型在诗、词、古文三种文体上自动断句效果最好，分别达到了99%、95%、92%的F1值。

以上研究已经在自动断句任务上取得了较好的结果，但自动标点的效果还有待提升，并且对长文本的句读处理方案关注较少。受前人研究启发，本文试图将BERT模型用于繁体古文自动句读，但由于谷歌发布的中文BERT模型是基于简体现代汉语语料训练得到的，并不一定能够很好地表示古文语义，本文利用大规模繁体古文语料对BERT中文模型进行增量训练，使其得到更好的繁体古文语义表示，然后再进行自动断句和自动标点训练。在实际的生产环境下，很多需整理的古籍的篇幅都较长，本文改进了数据串行滑动窗口方式并提出数据并行滑动窗口方式，能够有效解决篇章级自动句读准确率低和效率低的问题。

## 3 模型构建

预训练语言模型BERT的使用包括增量训练和微调两个阶段，以下分别介绍BERT增量模型训练和自动句读标点模型设置。

### 3.1 增量训练BERT模型

BERT模型由谷歌2018年发布，是多层transformer结构。BERT具有强大的语义表示功能，与传统的静态词向量不用，BERT能根据上下文生成动态的词向量，即同一个词在不同语境中会有不同的向量表示。BERT的训练过程是无监督的，能够自动从大量无标注语料中学习到字词和句子的语义表示。

我们从不同渠道收集大量繁体古文语料，包括诗歌、小说、骈文、论文等各种各类文体，内容包含经史子集、佛经等，文献分布年代广泛，从先秦至清代。经人工清洗整理，最后得到了约10亿字的带标点繁体古文语料。统计本文整理的语料得到的繁体字表有7万左右（包括各类异体字、古今字），BERT中文模型(以下称$\text{BERT}_{\text{base}}$)有固定词表，其中仅包含7321个汉字，覆盖率只有十分之一，如果直接使用$\text{BERT}_{\text{base}}$会使得很多繁体字在任务过程中被替换成UNK，造成语义不完整影响自动句读任务的效果。所以本文利用整理得到的语料对12层$\text{BERT}_{\text{base}}$做增量训练，并更换词表。根据$\text{BERT}_{\text{base}}$模型预训练步骤将增量训练分为三个阶段，前两个阶段的sequence length设为128，第三阶段设为512，第一阶段和第三阶段训练200K步，第二阶段训

练500K步。第一阶段仅更新embedding层的参数；第二阶段更新模型所有参数，使模型学习语义表示；第三阶段更新所有参数，使模型能学到长距离信息。最后得到增量繁体古文BERT模型，以下称$\text{BERT}_{\text{guwen}}$。

### 3.2 实验设置

以下分别介绍本文的实验数据集和自动句读标点模型。

#### 3.2.1 数据集

本文以学衡网[0] 200本核心典籍和github[1]公开的全中华古诗词数据库中的311860首诗作为实验语料，两部分皆经过人工整理，都是繁体汉字，且标点质量比较高。语料具体统计信息如Table 1所示，虽然最大句长超过万字，但统计发现97%的句长都在200字以内，句长分布如Figure 1所示。我们将数据集按照句子数8:1:1切分为训练集、测试集和验证集。为了让模型能处理较长文本，我们随机将同一段落中的3-10个句子合并作为一条训练数据。如表Table 2 ,Table Table 3所示的标注实例，本文选用二元标签BM进行断句数据标注，在二元标签基础上设计断句和标点联合标注标签。“B”表示对应的字符在句首，“M”表示对应字符在句中或句尾。‘D’、‘J’、‘Dun’、‘F’、‘M’、‘W’、‘G’分别表示该句以逗号、句号、顿号、分号、冒号、问号、感叹号结尾。

Table 1: 数据集统计信息

| | 句子数 | 字符数 | 最大句长 |
|---|---|---|---|
| 核心典籍 | 1161335 | 48887994 | 25339 |
| 诗 | 311861 | 16228612 | 2449 |
| 总计 | 1473196 | 65116606 | 25339 |

Table 2: 断句标注示例

| 庶 | 見 | 素 | 冠 | 兮 | 棘 | 人 | 欒 | 欒 | 兮 | 勞 | 心 | 慱 | 慱 | 兮 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B | M | M | M | M | B | M | M | M | M | B | M | M | M | M |

Table 3: 标点标注示例

| 庶 | 見 | 素 | 冠 | 兮 | 棘 | 人 | 欒 | 欒 | 兮 | 勞 | 心 | 慱 | 慱 | 兮 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B_D | M_D | M_D | M_D | M_D | B_D | M_D | M_D | M_D | M_D | B_J | M_J | M_J | M_J | M_J |

#### 3.2.2 自动句读模型

预训练模型可以通过微调迭代被调整为适合当前任务的模型，本文将自动句读和标点当作是预训练模型下游的序列标注任务。

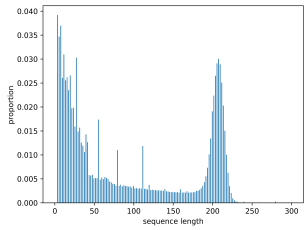



Figure 1: 数据集句长分布

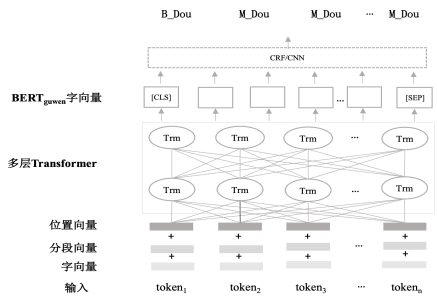



Figure 2: $\text{BERT}_{\text{guwen}}$+CRF/CNN 模型图

[0]http://core.xueheng.net/
[1]https://github.com/jackeyGao/chinese-poetry

2001年Lafferty等人提出的条件随机场模型(Conditional Random Field, CRF)，是一种无向图模型。该模型结合了最大熵和隐马尔可夫的特点，在词性标注、命名实体识别等序列标注任务中表现优异。虽然现在深度学习模型也可以很好的解决序列标注问题，但是增加CRF作为解码层似乎效果更好。如Huang等(2015)在做命名实体识别任务时考虑到标签前后的依赖性，在Bi-LSTM后接CRF层作为输出层，发现增加CRF层会比单独使用深度学习模型效果更好。本文也将CRF作为模型的最后一层，通过其学习标签之间的关系，找到全局最优的标签序列。

卷积神经网络(Convolutional Neural Networks, CNN)是一种前馈神经网络，可以在大量数据中识别序列的局部特征，并将它们生成为固定大小的向量表示，捕捉对当前任务最有效的特征。我们在$\text{BERT}_{\text{guwen}}$后接CNN层和全连接层，CNN能够在$\text{BERT}_{\text{guwen}}$的基础上对句子的上下文做进一步编码，捕捉局部特征。$\text{BERT}_{\text{guwen}}$+CRF/CNN模型如Figure 2所示。

BiLSTM-CRF模型是非常经典的序列标注模型(Huang et al., 2015)，我们将该模型作为基准模型，将BiLSTM的隐藏元数量设为256，词向量维度设为300。本文比较$\text{BERT}_{\text{guwen}}$+CRF、$\text{BERT}_{\text{base}}$+CRF、$\text{BERT}_{\text{guwen}}$+CNN、$\text{BERT}_{\text{base}}$+CNN模型在句读和标点任务上的表现，sequence length设为300，batch size设为32。实验在2块32G的Tesla V100 GPU上进行，每个模型训练到收敛为止。

## 4 实验结果及分析

### 4.1 断句实验结果

为检验不同模型在断句任务上的性能，本文使用精确率(precision)、召回率(recall)和F1(F1-score)作为评价指标。

断句实验结果如Table 4所示，可以看出诗歌断句结果整体好于古文断句结果，可能是因为古诗具有特定的体制和韵律，如五言绝句、七言律诗等，模型更加容易学得其断句规律，可以看到古诗断句最好的F1值已经超过99%。而古文的形式更加灵活，句式更加丰富，最好的断句F1为95.03%，比古诗低了4.5%。

对比不同模型的性能，可以看到$\text{BERT}_{\text{guwen}}$+CRF模型相比其他模型在断句任务上有最高的召回率和F1值，分别为95.16%、95.03%，$\text{BERT}_{\text{guwen}}$+CNN模型有最高的召回率95.13%。相比于基线模型Bi-LSTM+CRF，融入了预训练语言模型的模型断句效果均有一定程度的提升。融合增量训练的古文预训练模型的$\text{BERT}_{\text{guwen}}$+CRF模型相比于基线模型，F1值提高了6.55%。

Table 4: 断句实验结果

| | 古文 | | | 诗 | | |
|---|---|---|---|---|---|---|
| **Model** | *Precision* | *Recall* | *F1* | *Precision* | *Recall* | *F1* |
| **Bi-lstm+CRF** | 0.8441 | 0.7981 | 0.8205 | 0.9692 | 0.9660 | 0.9676 |
| ***BERT (俞敬松et al., 2019)** | 0.9232 | 0.9107 | 0.9167 | - | - | - |
| ***BERT+CNN (胡韧奋et al., 2019)** | 0.9177 | 0.9241 | 0.9203 | 0.9904 | 0.9935 | 0.9919 |
| **$\text{BERT}_{\text{base}}$+CNN** | 0.9296 | 0.9311 | 0.9303 | 0.9885 | 0.9914 | 0.9900 |
| **$\text{BERT}_{\text{base}}$+CRF** | 0.9301 | 0.9339 | 0.9320 | 0.9890 | 0.9923 | 0.9906 |
| **$\text{BERT}_{\text{guwen}}$+CNN** | **0.9513** | 0.9462 | 0.9487 | 0.9932 | 0.9950 | 0.9941 |
| **$\text{BERT}_{\text{guwen}}$+CRF** | 0.9490 | **0.9516** | **0.9503** | **0.9940** | **0.9967** | **0.9953** |

对比$\text{BERT}_{\text{base}}$+CRF和$\text{BERT}_{\text{guwen}}$+CRF的实验结果，可以看出使用了$\text{BERT}_{\text{guwen}}$的模型断句效果比使用$\text{BERT}_{\text{base}}$的模型好，F1值提高了1.99%，这说明对BERT模型做繁体古文增量训练，可能使模型学习到更多古文知识，能更好地处理断句任务。如以下案例所示，“用兵”其主语本是“朝廷”，在此处承前省略主语，“其主”与“秉常”属于同位语，共同作为“囚”的宾语，$\text{BERT}_{\text{guwen}}$+CRF经过了古文增量训练，能够更好地识别此类主语省略的句式，断句结果正确。而$\text{BERT}_{\text{base}}$+CRF模型错误地将“秉常”当作“用兵”的主语，“西方”作为“既下”的主语，导致断句错误。“城砦”为双音节文言词，在古文中属于比较常用的词，但在现代汉语中几乎不再使用，$\text{BERT}_{\text{base}}$+CRF不能准确的识别这一词语，可能因为在其现代汉语训练语料中“城砦”很少出现，$\text{BERT}_{\text{guwen}}$+CRF将“城砦”作为一个整体且断句正确，这说明增量训练之

后的$\text{BERT}_{\text{guwen}}$+CRF对文言词更加敏感。

原文：朝廷以夏人囚廢其主秉常。用兵西方。既下米脂等城砦數十。

$\text{BERT}_{\text{guwen}}$+CRF：朝廷以夏人囚廢其主秉常。用兵西方。既下米脂等城砦數十。

$\text{BERT}_{\text{base}}$+CRF：朝廷以夏人囚廢其主。秉常用兵。西方既下。米脂等城。砦數十。

通过分析断句结果，我们发现断句经常出现“可断可不断”的情况，如以下两个案例所示，原文为“借兵於楚伐魏”，模型断句结果为“借兵於楚。伐魏”，在“伐魏”之前断句应该也不为错误。案例2的模型断句也是类似的情况，模型断句偏向于将长句断为小句，但这种断句结果似乎不能算作错误。在实验时，将唯一断句标注集作为标准答案，并不能全面地评估模型的性能，以后可以尝试在测试集中给出多种正确标注答案。

$\text{BERT}_{\text{guwen}}$+CRF结果

案例1：

原文：取我剛平。六年。借兵於楚伐魏。

$\text{BERT}_{\text{guwen}}$+CRF：取我剛平。六年。借兵於楚。伐魏。

案例2：

原文故曰。禮人而不荅則反其敬。愛人而不親則反其仁。治人而不治則反其知。

$\text{BERT}_{\text{guwen}}$+CRF：故曰。禮人而不荅。則反其敬。愛人而不親。則反其仁。治人而不治。則反其知。

### 4.2 标点实验结果

本文在评价标点模型时使用微平均精确率($\text{P}_{\text{micro}}$)、召回率($\text{R}_{\text{micro}}$)和($\text{F1}_{\text{micro}}$)。标点实验结果如Table 5所示，由于诗歌的标点规则比较简单，所有模型的标点F1值都在95%以上。$\text{BERT}_{\text{guwen}}$+CNN模型在古文和诗歌上标点表现最好，F1值为80.18%。$\text{BERT}_{\text{guwen}}$+CRF比$\text{BERT}_{\text{base}}$+CRF的标点F1值高1.67%，$\text{BERT}_{\text{guwen}}$+CNN比$\text{BERT}_{\text{base}}$+CNN的标点F1值高2.81%，说明增量训练之后的模型在一定程度上能够帮助提升标点效果。

和断句任务的结果相比，标点的准确率、召回率、F1值与断句均有较大差距，因为断句规则相对比较统一，而标点的规则比较复杂，不同的标点表达不同的感情和意义。我们实验的语料虽然是经过人工整理的，但是依然存在标注规则不一致的情况，如逗号和句号、分号和逗号的使用常常因人而异，模型也难以分辨。

Table 5: 标点实验结果

| | 古文 | | | 诗 | | |
|---|---|---|---|---|---|---|
| **Model** | *Precision* | *Recall* | *F1* | *Precision* | *Recall* | *F1* |
| **Bi-lstm+CRF** | 0.5681 | 0.5149 | 0.5402 | 0.9646 | 0.9614 | 0.9630 |
| ***BERT (俞敬松et al., 2019)** | 0.7092 | 0.6988 | 0.7040 | - | - | - |
| **$\text{BERT}_{\text{base}}$+CNN** | 0.7831 | 0.7763 | 0.7797 | 0.9883 | 0.9867 | 0.9875 |
| **$\text{BERT}_{\text{base}}$+CRF** | 0.7827 | 0.7775 | 0.7801 | 0.9734 | 0.9603 | 0.9668 |
| **$\text{BERT}_{\text{guwen}}$+CNN** | 0.7954 | **0.8084** | **0.8018** | **0.9893** | **0.9890** | **0.9891** |
| **$\text{BERT}_{\text{guwen}}$+CRF** | **0.7943** | 0.7966 | 0.7955 | 0.9887 | 0.9870 | 0.9879 |

### 4.3 增量古文模型语义表示能力

上面的实验结果已经证明$\text{BERT}_{\text{guwen}}$模型比$\text{BERT}_{\text{base}}$模型在断句和标点任务上表现更好。本文设计实验进一步讨论$\text{BERT}_{\text{guwen}}$的表现优于谷歌的$\text{BERT}_{\text{base}}$的原因。

古代汉语和现代汉语各有特点，现代汉语以双音节词为主，古代汉语以单音节词为主，且多义词比例很高。BERT与传统的词向量模型不同，BERT能够对不同语境下同一个词有不同的语义表示，具有区分同一个词的不同义项的能力，如“君之病在肠胃”中的“病”与“人皆嗤吾固陋，吾不以为病”中的“病”分别对应不同的向量。

本文选取一组古汉语多义词来讨论$\text{BERT}_{\text{guwen}}$和$\text{BERT}_{\text{base}}$对文言词的语义表示能力。本文选取古汉语多义词基于以下三个原则：1）单音节多义词，因为BERT中文模型只能对句子和单

字词做语义表示；2）词语义项多，古言词除本义外通常还有引申义和假借义；3）词语在古汉语中使用率高，属于常用词。

基于以上三点，我们参考文学网[2]发布的150个古文多义实词以及《古汉语常用字字典》第四版，选取“安”、“謝”、“信”、“兵”、“愛”、“病”、“假”七个单音节词作为实验对象，以上七个多义词义项都在3个以上，并且在我们的语料中出现频次高于500万次。

首先从整理的语料中分别找到3000条含有以上七个单音节词的句子，利用$\text{BERT}_{\text{guwen}}$对每条例句中的词作向量化表示，然后用k-means对以上七个词语的所有词向量作聚类，最后使用t-nse对聚类结果进行可视化。根据《古汉语常用字字典》中的义项，七个单字词的义项共36个，将k-means的聚类数设为36，模型自动将所有词向量聚为36个小类。聚类效果如图3所示，图中每个点代表一个词向量，从Figure 3上可以比较明显地看出聚类之后出现了七个模块，每一模块对应一个文言单字词，每个模块内部又包含不同颜色的点，不同颜色表示词内部有不同的义项。以上聚类结果说明$\text{BERT}_{\text{guwen}}$能够将不同文言词的语义区分开，并且能表示出一个多义词的不同义项。

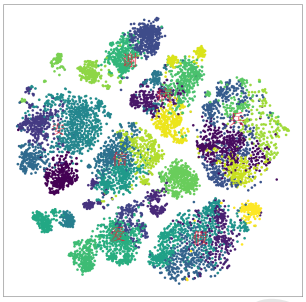


Figure 3: k-means对七个古汉语单字词向量的聚类效果图

为了进一步讨论$\text{BERT}_{\text{guwen}}$模型对同一个文言词的不同义项的区分能力，我们对比$\text{BERT}_{\text{guwen}}$和$\text{BERT}_{\text{base}}$两个模型对七个多义词的不同义项的语义表示能力，即是否能将不同义项分开。以“安”和“謝”为例，首先根据文言词“安”的四个常用义项人工挑出2000条例句，根据文言词“谢”的三个常用义项挑出1500条例句，部分例句如Table 6所示。分别使用$\text{BERT}_{\text{guwen}}$和$\text{BERT}_{\text{base}}$两个模型生成“安”和“谢”在所有例句中的词向量，最后进行聚类。我们使用轮廓系数评估聚类结果，聚类效果越好轮廓系数越高，计算公式如下：

$$a(i) = \frac{1}{n-1}\sum_{j\neq i}^{n} distance(i,j) \tag{1}$$

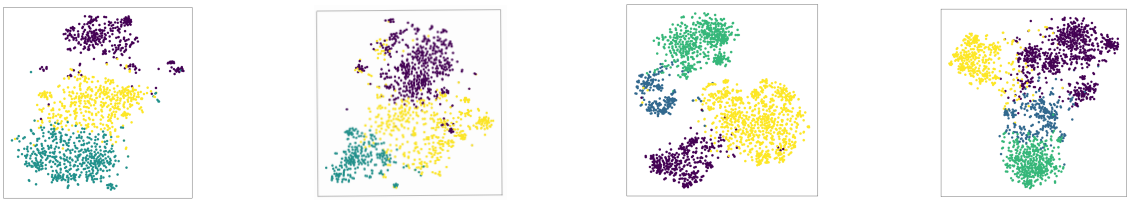
$a(i)$表示样本点$i$的簇内不相似度，$j$表示与样本$i$在同一个类中的其他样本，$distance(i,j)$表示$i$和$j$之间的距离。

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2}$$

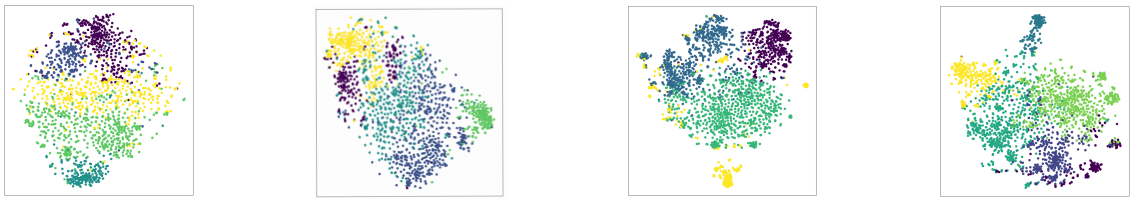
$b(i)$表示$i$和其他每个类别的所有样本之间的距离和的最小值，计算方式和$a(i)$类似。所有样本的$S(i)$均值即为聚类结果的轮廓系数。

如Figure 4所示，Figure 4(a)为$\text{BERT}_{\text{guwen}}$ 生成的“謝”的词向量的聚类效果,聚类系数为3，轮廓系数S为0.1173；Figure 4(b)为$\text{BERT}_{\text{base}}$ 生成“謝”的词向量的聚类效果,聚类系数为3，轮廓系数S为0.0964；对比图Figure 4(a)和Figure 4(b)，发现$\text{BERT}_{\text{guwen}}$生成的“謝”的向量能够被清晰地聚为3类，且Figure 4(a)的轮廓系大于Figure 4(b)的轮廓系数。对比七个多义词的七组聚类效果图及其轮廓系数，发现除了“信”以外，$\text{BERT}_{\text{guwen}}$生成的词向量的聚类效果明显好于$\text{BERT}_{\text{base}}$生成的词向量。

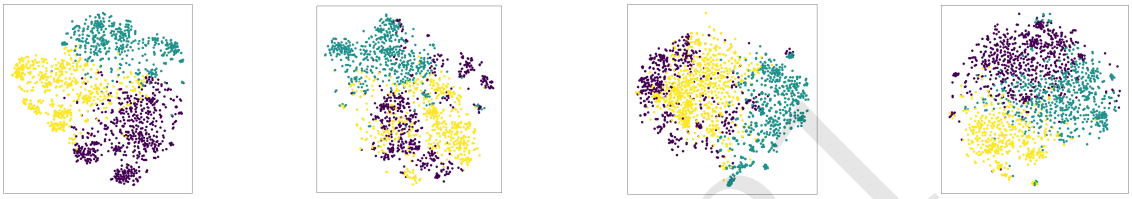
[2]https://wyw.hwxnet.com/article/24.html



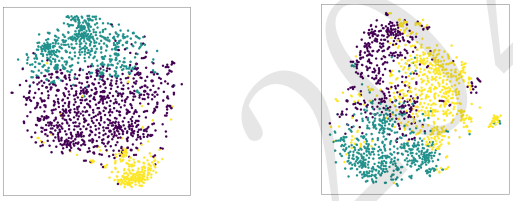
(a) $\text{BERT}_{\text{guwen}}$謝,S=0.1173 (b) $\text{BERT}_{\text{base}}$,謝,S=0.0964 (c) $\text{BERT}_{\text{guwen}}$,安,S=0.1365 (d) $\text{BERT}_{\text{base}}$,安,S= 0.1275



(e) $\text{BERT}_{\text{guwen}}$,假,S=0.1015 (f) $\text{BERT}_{\text{base}}$,假,S=0.0971 (g) $\text{BERT}_{\text{guwen}}$,信,S=0.0625 (h) $\text{BERT}_{\text{base}}$信,S=0.0795



(i) $\text{BERT}_{\text{guwen}}$,愛,S=0.0754 (j) $\text{BERT}_{\text{base}}$,愛, S=0.0629 (k) $\text{BERT}_{\text{guwen}}$,病,S=0.0703 (l) $\text{BERT}_{\text{base}}$,病,S=0.0676



(m) $\text{BERT}_{\text{guwen}}$,兵,S=0.0713 (n) $\text{BERT}_{\text{base}}$,兵,S=0.0671

Figure 4: 七组多义词由$\text{BERT}_{\text{guwen}}$和$\text{BERT}_{\text{base}}$生成的词向量的聚类效果图，S表示聚类轮廓系数。

Table 6: 文言词“安”、“谢”常用义项例句（部分）

| 安 | 例句 | 謝 | 例句 |
|---|---|---|---|
| 疑问代词：哪里，什么地方 | 沛公【安】在<br>元元【安】所歸命哉<br>其符【安】在<br>皮之不存，毛將【安】傅 | 动词：感谢 | 噲拜【謝】，起，立而飲之<br>安世嘗有所薦，其人來【謝】 |
| 疑问副词：怎么 | 燕雀【安】知鴻鵠之誌哉<br>【安】能辨我是雌雄<br>【安】肯降之乎<br>【安】有伯夷、叔齊 | 动词：道歉 | 入而徐趨，至而自【謝】<br>旦日不可不蚤自來【謝】項王<br>因賓客至蘭相如門【謝】罪 |
| 形容词：安逸、安稳、安宁 | 世書俗説，多所不【安】<br>喪亂既平，既【安】且寧<br>【安】坐不動；或入瓶内 | 动词：凋谢 | 形【謝】而神滅<br>落魄東風不借春，吹開吹【謝】兩何因<br>不用鏡前空有淚，薔薇花【謝】即歸來 |
| 动词：安置、使...安 | 軍民【安】泰<br>京城家裏大小平【安】 | | |

观察“信”的聚类效果图，我们可以看出$\text{BERT}_{\text{guwen}}$的聚类效果似乎好于$\text{BERT}_{\text{base}}$，但轮廓

系数前者却小于后者。原因可能是“信”的义项较多，并且这些义项之间本身有比较紧密的引申关系，词性主要是动词和名词。如“不欺，讲信用”（言而有信）、“信任”（愿陛下亲之信之）、“相信“（忌不自信）、“信用”（小信未孚，神弗福也）。而如“安”、“谢”这类多义词，不同义项距离较远，且词性多样。

## 5 篇章级断句

近年来，不断有学者提出长文本处理模型，BlockBERT(Qiu et al., 2020)切断BERT中不重要的注意力头，将BERT可处理的token数从512个扩展大到1024个。Big bird(Zaheer et al., 2020)模型使用稀疏注意力机制，将计算复杂度降到线性，可以处理比全局注意力transformer长8倍的序列。但是这类模型能处理的长度依然有限，长文本句读是生产环境下需要解决的问题，但目前涉及这一问题的研究较少。俞敬松等 (2019)使用滑动窗口的方式处理篇章级句读，如以下示例所示，每次输入不超过64字的片段，因其训练数据最长为21字，所以只取输出结果的前一个或两个断句结果，剩余的部分归并到第二次切分的64字。这种滑动窗口方式虽然在一定程度上保证了断句的准确性，但是每次处理的序列只有64字，且每次只取前两句的断句结果，后面的处理结果因准确性不高都被放弃。这种方式每次需等待前一片段输出结果之后才能进行第二片段的处理，处理效率很低。

串行滑动窗口1示例:

第一次输入64字

青州黃巾衆百萬入兗州殺任城相鄭遂轉入東平劉岱欲擊之鮑信諫曰..... 略為資今不若.....

第二次输入64字

青州黃巾衆百萬入兗州。殺任城相鄭遂。轉入東平。劉岱欲擊之。鮑信諫曰。..... 略為資今不若.....

第一次输出结果

第一次取得结果：青州黃巾衆百萬入兗州。殺任城相鄭遂。

本文提出了两种新的滑动窗口方式，在保证准确率的同时也能极大提高运行速率，以下称串行滑动窗口2和并行滑动窗口。串行滑动窗口2是通过对串行滑动窗口1改进得到，如以下示例，首先输入文档的前125个字，然后等模型返回前125个字的断句结果，因为倒数第一句可能因为语义不完整而出现错误断句，所以将倒数第一句的断句结果加入到下一次切分的125字中，依次处理完整片文本。这种方法使得每次能处理更长的序列，并且每次只放弃输出结果的最后一句，运行速度相比串行滑动窗口1有一定提高。但是因为数据处理的方式仍然是串行的，每次需要等待前面的返回结果，句读效率不足以满足使用需求。

串行滑动窗口2示例:

第一次输入125字

.....據其要害擊之可破也岱不從遂與戰果為所殺.....遂進兵擊黃巾於壽張東信力戰死.....

第二次输入125字

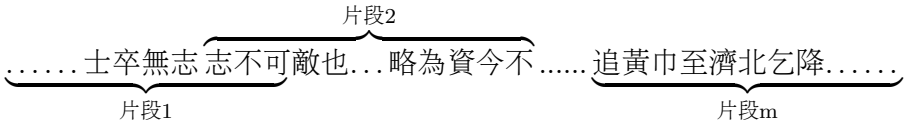
.....據其要害。擊之可破也。岱不從。遂與戰。果為所殺。.....遂進兵擊黃巾於壽張東信力戰死.....

第一次输出结果

第一次取得结果：.....據其要害。擊之可破也。岱不從。遂與戰。

为了进一步提高篇章级句读速率，本文提出了并行滑动窗口方法。如以下案例所示，将长文本数据按照滑动窗口的方式切分，第一个片段与第二个片段重复n个字，第三个片段与第二个片段重复n个字，依次将长文本切成m个片段，将m个片段同时送入模型，同时返回m个结果。在处理返回结果时也按照滑动窗口的方式处理，对于片段1，首先删除倒数第一句的输出结果得到新的片段1，然后在片段2的输出结果中删除和新的片段1重复的部分，同样删除片段2的倒数第一句的输出结果，得到新的片段2，最后将新的片段1和新的片段2拼接，依次将所有的片段拼接得到最后的输出序列。将一整篇文本切分为多条数据并行处理，大幅度提高了句读速度，并且能保证句读的准确率。在实验中我们将片段长度设置为125，重复字数n设为20。

并行滑动窗口示例:




片段1句读结果：.....士卒無志。不可敵。

新的片段1

片段2句读结果：志。不可敵也。...略為資。今不。

新的片段2

句读最终结果：.....士卒無志。不可敵也。...略為資。.....

新的片段1　新的片段2

我们将直接截断的方式作为基线标准，将长文本每64字截断组成一批数据喂进模型。使用以上四种方式句读一段4168字的长文本，实验结果如Table 7所示。

从Table 7中可以比较明显地看出，滑动窗口方法的F1值都高于直接截断的方法，这是因为滑动窗口只取语义比较完整的文本片段作为输出结果，而直接截断的方式容易造成文本片段结尾强制断句的错误，但是直接截断的方式具有最高的处理效率。对比两种串行滑动窗口方式，本文改进的串行滑动窗口2句读速度相比于串行滑动窗口1提高了11倍，且有最高的F1值。比较并行滑动窗口和两种串行滑动窗口，并行滑动窗口方式用时5.79s，和直接截断方式用时基本无差，同时也保证了断句具有较高的F1值和准确率。

Table 7: 四种篇章级句读方法实验结果。window_size表示滑动窗口大小，即切分的片段的长度。Time-using指处理完整个文本所用时间，以秒(s)为单位。

| | 直接截断 | 串行滑动窗口1 (2019) | 串行滑动窗口2 | 并行滑动窗口 |
|---|---|---|---|---|
| **Window_size** | 64 | 64 | 128 | 128 |
| **Precision** | 0.8657 | 0.8931 | 0.9207 | **0.9242** |
| **Recall** | 0.9522 | 0.9597 | **0.9701** | 0.9642 |
| **F1** | 0.9069 | 0.9252 | **0.9448** | 0.9438 |
| **Time-using** | **5.74s** | 1288.58s | 115.28s | 5.79s |

基于本文提出的句读模型和并行滑动窗口方式，我们开发了“吾与点”古籍自动句读平台(https://wyd.kvlab.org/)。该平台可以帮助古籍研究者和爱好者自动句读古籍文本。

## 6 总结

古文断句和标点是古籍整理过程中重要的一步，本文利用预训练语言模型实现了繁体古籍的自动断句和标点。首先利用10亿字繁体古文语料对谷歌中文BERT模型做增量训练，然后以此预训练模型为基础实现了繁体古文的自动断句和标点。古文和诗歌的自动断句F1值分别为95.03%，99.53%，标点F1值分别为80.18%，98.81%。并且我们发现增量训练后的BERT模型能够提升自动断句和自动标点的效果。本文通过对文言多义词的多个义项聚类发现增量训练的语言模型的古文语义表示能力优于谷歌原始BERT模型，并且具备一定的区分多义词不同义项的能力。在篇章级句读方面，本文改进了数据串行方案并提出数据并行滑动窗口方式，既能保证句读的准确率也能保持极高的处理效率。

## 参考文献

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Z. Huang, X. Wei, and Y. Kai. 2015. Bidirectional lstm-crf models for sequence tagging. *Computer Science*.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *https://arxiv.org/pdf/1907.11692.pdf*.

J. Qiu, H. Ma, O. Levy, W. T. Yih, and J. Tang. 2020. Blockwise self-attention for long document understanding. In *Findings of EMNLP'20*.

M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, and L. Yang. 2020. Big bird: Transformers for longer sequences. *https://arxiv.org/abs/2007.14062*.

俞敬松, 魏一, and 张永伟. 2019. 基于bert的古文断句研究与应用. 中文信息学报, 33(11).

张开旭, 夏云庆, and 宇航. 2009. 基于条件随机场的古汉语自动断句与标点方法. 清华大学学报:自然科学版.

王博立, 史晓东, and 苏劲松. 2017. 一种基于循环神经网络的古文断句方法. 北京大学学报(自然科学版), 53(02):255–261.

程宁, 李斌, 葛四嘉, 郝星月, and 冯敏萱. 2020. 基于bilstm-crf的古汉语自动断句与词法分析一体化研究. 中文信息学报, 034(004):1–9.

胡韧奋, 李绅, and 诸雨辰. 2019. 基于深层语言模型的古汉语知识表示及自动断句研究. In 第十八届中国计算语言学大会.

释贤超. 2018. 自动标点的原理与实现. In 第9届数位典藏与数位人文国际研讨会.

陈天莹, 陈蓉, 潘璐璐, 李红军, and 于中华. 2007. 基于前后文n-gram模型的古汉语句子切分. 计算机工程.

黄建年and 侯汉清. 2008. 农业古籍断句标点模式研究. 中文信息学报, 022(004):31–38.

黄水清and 王东波. 2017. 古文信息处理研究的现状及趋势. 图书情报工作, 12(v.39;No.267):44–50.

黄瀚萱. 2008. 以序列标注方法解决古汉语断句问题. Ph.D. thesis, 台湾交通大学.