（命名）实体识别NER已有古文预训练模型测评实验报告

**数据集**

数据集A取自《新五代史》《北史》等古籍，训练集，验证集和测试集分别为21128，2831，2148，平均句长为26，采用BIO，共有5种实体类型，[O，B-NOUN\_BOOKNAME，I-NOUN\_BOOKNAME，B-NOUN-OTHER ，I-NOUN-OTHER]

数据集B取自《春秋谷梁传》等古籍，训练集，验证集和测试集分别为7528，865，877，平均字长为85，采用BIO，共有13 种实体类型，[O ,B-PER ,I-PER, B-ORG ,I-ORG ,B-LOC ,I-LOC, B-JOB ,I-JOB ,B-WAR, I-WAR, B-BOO, I-BOO]

实验数据下载：

https://github.com/jizijing/C-CLUE

<https://github.com/Ethan-yt/CCLUE>

**评价指标**

**准确率(Accuracy)**

Accuracy是从整体上衡量模型的性能，即模型预测正确的样本占全部样本的比例。它的计算公式如下：



其中，TP（True Positive）表示真正例，即模型预测为正例且实际上也为正例的样本数量；FP（False Positive）表示假正例，即模型预测为正例但实际上为负例的样本数量；TN（True Negative）表示真负例，即模型预测为负例且实际上也为负例的样本数量；FN（False Negative）表示假负例，即模型预测为负例但实际上为正例的样本数量。

**精确率(Precision)**

Precision关注的是模型预测为正例的样本中有多少是真正的正例。换句话说，它衡量的是模型预测的正例中有多少是正确的，计算公式为：



其中，TP（True Positive）表示真正例，即模型预测为正例且实际上也为正例的样本数量；FP（False Positive）表示假正例，即模型预测为正例但实际上为负例的样本数量。

**F1值(F1 Score)**

F1分数是精确率和召回率的调和平均值，用于衡量模型的准确性。它的计算公式如下：



其中，精确率（Precision）是指模型预测为正例的样本中真正为正例的比例，召回率（Recall）是指所有真正的正例样本中被模型预测为正例的比例。

**召回率(Recall)**

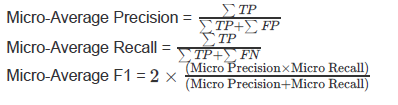
召回率是指在所有真正的正例样本中，被模型正确预测为正例的比例。它的计算公式如下：



其中，TP（True Positive）表示真正例，即模型预测为正例且实际上也为正例的样本数量；FN（False Negative）表示假负例，即模型预测为负例但实际上为正例的样本数量。

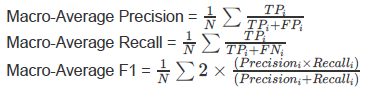
**Micro-Average**

Micro-Average 把所有类别的结果汇总起来计算平均值。它将所有类别的贡献视为等同，因此对于样本量大的类别，Micro-Average 更加敏感。



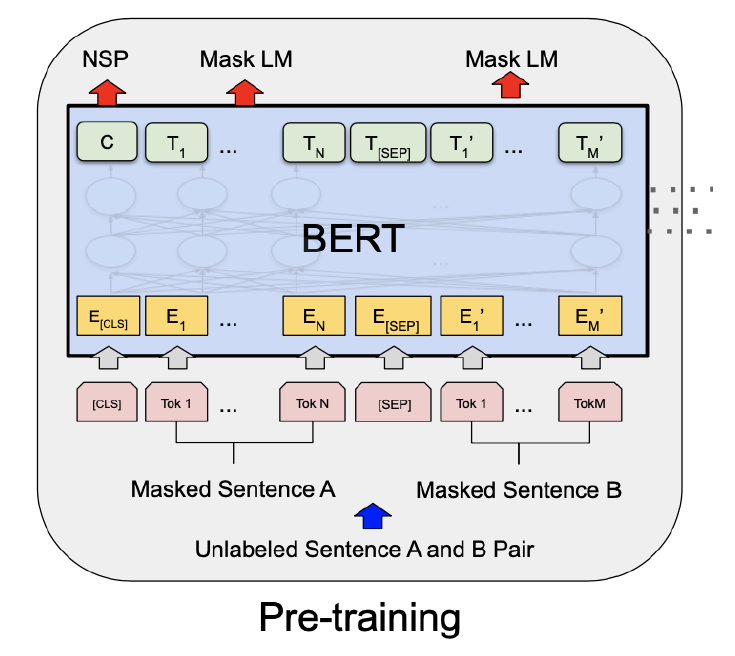
**Macro-Average**

Macro-Average 分别计算每个类别的性能指标，然后计算这些指标的算术平均值。它给予每个类别同等的权重，无论类别的样本量大小，因此对于样本量小的类别更加敏感。



**模型**

**BERT**



BertModel 部分包含了BERT的所有主要组件，包括词嵌入层、位置嵌入层、令牌类型嵌入层、层归一化层和Dropout层。

BertEncoder 部分负责处理输入序列，通过多个BERT层来进行特征提取。每个BERT层都包含注意力机制和前馈神经网络。

BertForTokenClassification 模型的最后一部分是一个线性分类器，它将BERT编码器的输出映射到目标类别的概率分布上。

**1.Transformer架构：** BERT建立在Transformer模型的基础上，这是一种使用自注意力机制（Self-Attention Mechanism）的深度神经网络。Transformer允许模型在处理序列数据时同时关注序列中的所有位置，而不是像传统的循环神经网络（RNN）或卷积神经网络（CNN）那样逐步处理。

**2.预训练策略：** BERT采用了无监督的预训练策略，通过大规模的语言模型预训练来学习丰富的语义表示。该模型通过对大量文本数据进行“遮蔽语言模型”（Masked Language Model，MLM）任务的预训练，使得模型能够理解词汇和语法结构，并捕捉单词之间的关系。

**3.双向性：** BERT在预训练时考虑了双向信息，即使用上下文信息来理解每个词的语义。这种双向性有助于模型更好地理解文本中的语境和关联，提高了对上下文相关性的捕捉。

**4.Fine-tuning：** 预训练后，BERT模型可以通过微调（fine-tuning）来适应特定的下游任务，如命名实体识别、情感分析等。这种能力使得BERT在各种NLP任务中都表现出色，无需从零开始训练新的模型。

**5.Contextual Embeddings：** BERT生成的词向量是上下文相关的，每个词的表示取决于整个输入句子的上下文，而不是简单地从固定的嵌入中获取。这种上下文敏感的嵌入有助于更准确地捕捉语义信息。

**实验结果**

**Bert-ancient-Chinese**

**训练最终结果**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **数据集** | **Eval Precision** | **Eval Recall** | **Eval F1** | **Eval Accuracy** | **Eval Loss** | **Eval Runtime** |
| **数据集A** | **0.9713613930185667** | **0.9717125382262997** | **0.9715153155843004** | **0.9717125382262997** | **0.1591310352087021** | **4.3364** |
| **数据集B** | **0.9607732624971154** | **0.9653663548752834** | **0.9627669359301028** | **0.9653663548752834** | **0.17902402579784393** | **8.1976** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **数据集** | **Micro-Average Precision** | **Micro-Average Recall** | **Micro-Average F1** | **Macro-Average Precision** | **Macro-Average Recall** | **Macro-Average F1** |
| **数据集A** | **0.9717125382262997** | **0.9717125382262997** | **0.9717125382262997** | **0.8673289276579453** | **0.8478130526942544** | **0.8573962308701384** |
| **数据集B** | **0.9911334325396826** | **0.9911334325396826** | **0.9911334325396826** | **0.24364633330184382** | **0.21699121039347588** | **0.21620752290443754** |

**模型介绍和分析**

这是一个基于BERT的模型，专门训练用于古汉语处理。它适用于古文的语义理解和文本生成，旨在提高对古汉语的处理能力。

**训练过程分析**

**数据集A**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Eval Precision** | **Eval Recall** | **Eval F1** | **Eval Accuracy** | **Eval Loss** | **Eval Runtime** |
| **2** | **0.965859360425647** | **0.9673042704626335** | **0.9661755533216788** | **0.9673042704626335** | **0.10091111063957214** | **3.5747** |
| **4** | **0.9695756273928118** | **0.9709185943060499** | **0.9698405783513404** | **0.9709185943060499** | **0.09781475365161896** | **3.5827** |
| **6** | **0.9748397118458895** | **0.9753113879003559** | **0.9749971259961178** | **0.9753113879003559** | **0.10068126767873764** | **3.5825** |
| **8** | **0.9729364759002894** | **0.9735042259786477** | **0.9731642014794695** | **0.9735042259786477** | **0.11649686098098755** | **3.5887** |
| **10** | **0.9743676182461154** | **0.9747553380782918** | **0.974540298619832** | **0.9747553380782918** | **0.13155880570411682** | **3.5785** |

**数据集B**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Eval Precision** | **Eval Recall** | **Eval F1** | **Eval Accuracy** | **Eval Loss** | **Eval Runtime** |
| **2** | **0.9634874417593906** | **0.9647728883219955** | **0.9636102287355702** | **0.9647728883219955** | **0.08292320370674133** | **7.6235** |
| **4** | **0.9611707205126476** | **0.9681919642857143** | **0.9634537970119926** | **0.9681919642857143** | **0.08083999902009964** | **7.7177** |
| **6** | **0.9619109614420617** | **0.9659775368480725** | **0.9631737884030217** | **0.9659775368480725** | **0.10151440650224686** | **7.5857** |
| **8** | **0.9607420133460248** | **0.9666772959183674** | **0.9630476118906922** | **0.9666772959183674** | **0.13081666827201843** | **7.5832** |
| **10** | **0.9608151673880866** | **0.9650031887755102** | **0.9626049702565307** | **0.9650031887755102** | **0.15603597462177277** | **7.5866** |

**SikuRoberta**

**训练最终结果**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **数据集** | **Eval Precision** | **Eval Recall** | **Eval F1** | **Eval Accuracy** | **Eval Loss** | **Eval Runtime** |
| **数据集A** | **0.9681120155776729** | **0.9683438455657493** | **0.9682108016177119** | **0.9683438455657493** | **0.15556995570659637** | **4.3383** |
| **数据集B** | **0.9617054936611624** | **0.9659598214285714** | **0.9635556888246802** | **0.9659598214285714** | **0.15483438968658447** | **8.1275** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **数据集** | **Micro-Average Precision** | **Micro-Average Recall** | **Micro-Average F1** | **Macro-Average Precision** | **Macro-Average Recall** | **Macro-Average F1** |
| **数据集A** | **0.9683438455657493** | **0.9683438455657493** | **0.9683438455657493** | **0.8362269475264886** | **0.8354067662968905** | **0.8356550520943984** |
| **数据集B** | **0.9911068594104309** | **0.9911068594104309** | **0.9911068594104309** | **0.23185115660763794** | **0.23071127190351204** | **0.22199941287605457** |

**模型介绍和分析**

这个模型是Siku Quanshu（四库全书）的基础上训练的RoBERTa变体。它对古典文献中的文本理解有很强的能力，适合处理古籍和文献分析任务。

**训练过程分析**

**数据集A**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Eval Precision** | **Eval Recall** | **Eval F1** | **Eval Accuracy** | **Eval Loss** | **Eval Runtime** |
| **2** | **0.9637076986681025** | **0.9639401690391459** | **0.9637398833761893** | **0.9639401690391459** | **0.11276831477880478** | **3.5975** |
| **4** | **0.9678154871484845** | **0.9688056049822064** | **0.9674825752554453** | **0.9688056049822064** | **0.09962229430675507** | **3.5792** |
| **6** | **0.9705103710023891** | **0.9706961743772242** | **0.9705351202725301** | **0.9706961743772242** | **0.11253400146961212** | **3.5679** |
| **8** | **0.9727463190910541** | **0.9733374110320284** | **0.9729854307448372** | **0.9733374110320284** | **0.12045649439096451** | **3.5799** |
| **10** | **0.9729097687048657** | **0.9735042259786477** | **0.9731459818475224** | **0.9735042259786477** | **0.13948172330856323** | **3.5832** |

**数据集B**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Eval Precision** | **Eval Recall** | **Eval F1** | **Eval Accuracy** | **Eval Loss** | **Eval Runtime** |
| **2** | **0.9612256192641184** | **0.9679350907029478** | **0.9630154282453001** | **0.9679350907029478** | **0.08013196289539337** | **7.4949** |
| **4** | **0.9612151459441082** | **0.9696534863945578** | **0.9622368460374825** | **0.9696534863945578** | **0.07940705120563507** | **7.7477** |
| **6** | **0.9617408635243314** | **0.9676959325396826** | **0.9637286930779716** | **0.9676959325396826** | **0.09067484736442566** | **7.796** |
| **8** | **0.9619194289796517** | **0.9665089994331065** | **0.9638604528771143** | **0.9665089994331065** | **0.11791561543941498** | **7.7083** |
| **10** | **0.9618047987592484** | **0.9661369756235828** | **0.9635584547151035** | **0.9661369756235828** | **0.13009293377399445** | **7.9148** |

**AnchiBERT**

**训练最终结果**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **数据集** | **Eval Precision** | **Eval Recall** | **Eval F1** | **Eval Accuracy** | **Eval Loss** | **Eval Runtime** |
| **数据集A** | **0.9639402429274961** | **0.9646167813455657** | **0.964243622737465** | **0.9646167813455657** | **0.1848205178976059** | **3.9459** |
| **数据集B** | **0.9614060718940829** | **0.9663495606575964** | **0.963474939687441** | **0.9663495606575964** | **0.16236752271652222** | **7.9751** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **数据集** | **Micro-Average Precision** | **Micro-Average Recall** | **Micro-Average F1** | **Macro-Average Precision** | **Macro-Average Recall** | **Macro-Average F1** |
| **数据集A** | **0.9646167813455657** | **0.9646167813455657** | **0.9646167813455657** | **0.8273717706255621** | **0.8038283103729272** | **0.8153108407512697** |
| **数据集B** | **0.9911068594104309** | **0.9911068594104309** | **0.9911068594104309** | **0.2512589632357055** | **0.24764968296968584** | **0.23697334205426301** |

**模型介绍和分析**

AnchiBERT是针对古代汉语的BERT变体，提供了对古汉语语料的更深入的理解。它优化了BERT架构以适应古汉语的特殊需求，适合古文翻译和解析。

**训练过程分析**

**数据集A**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Eval Precision** | **Eval Recall** | **Eval F1** | **Eval Accuracy** | **Eval Loss** | **Eval Runtime** |
| **2** | **0.9591849504729812** | **0.9614657473309609** | **0.9598301822708686** | **0.9614657473309609** | **0.1168164536356926** | **3.0897** |
| **4** | **0.9639877151221328** | **0.9652746886120996** | **0.9642676070302236** | **0.9652746886120996** | **0.11092515289783478** | **3.134** |
| **6** | **0.9666516636368039** | **0.9671930604982206** | **0.9668701972485794** | **0.9671930604982206** | **0.12030474841594696** | **3.1427** |
| **8** | **0.9662283521442985** | **0.9675266903914591** | **0.9666246275754931** | **0.9675266903914591** | **0.1335839033126831** | **3.0676** |
| **10** | **0.9687623978945292** | **0.9695006672597865** | **0.9690825948759366** | **0.9695006672597865** | **0.15416313707828522** | **3.0511** |

**数据集B**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Eval Precision** | **Eval Recall** | **Eval F1** | **Eval Accuracy** | **Eval Loss** | **Eval Runtime** |
| **2** | **0.9640783197007398** | **0.9652512046485261** | **0.9639040402066386** | **0.9652512046485261** | **0.08222125470638275** | **9.6945** |
| **4** | **0.9612196233374363** | **0.9690865929705216** | **0.9629341677537446** | **0.9690865929705216** | **0.08101007342338562** | **9.3474** |
| **6** | **0.9614725417305747** | **0.9673770549886621** | **0.963399964029979** | **0.9673770549886621** | **0.09246400743722916** | **7.1856** |
| **8** | **0.9612431850099584** | **0.9675896400226758** | **0.9636078031217096** | **0.9675896400226758** | **0.13486379384994507** | **6.7207** |
| **10** | **0.9614162998781131** | **0.9667127267573696** | **0.9635052248092323** | **0.9667127267573696** | **0.13726025819778442** | **6.6715** |

**guwenbert-base**

**训练最终结果**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **数据集** | **Eval Precision** | **Eval Recall** | **Eval F1** | **Eval Accuracy** | **Eval Loss** | **Eval Runtime** |
| **数据集A** | **0.8980535214526109** | **0.9162605122324159** | **0.900511550865519** | **0.9162605122324159** | **0.304275244474411** | **4.3513** |
| **数据集B** | **0.9501693842044238** | **0.9617346938775511** | **0.9549950423377727** | **0.9617346938775511** | **0.13864931464195251** | **8.1359** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **数据集** | **Micro-Average Precision** | **Micro-Average Recall** | **Micro-Average F1** | **Macro-Average Precision** | **Macro-Average Recall** | **Macro-Average F1** |
| **数据集A** | **0.9162605122324159** | **0.9162605122324159** | **0.9162605122324159** | **0.6532090679952062** | **0.396806130054585** | **0.46904291833949097** |
| **数据集B** | **0.9959254535147393** | **0.9959254535147393** | **0.9959254535147393** | **0.40479139253470026** | **0.4110982524602927** | **0.3914383970521158** |

**模型介绍和分析**

guwenbert-base是一个专门为古文设计的BERT模型。它利用大量古文语料进行训练，目标是提高古文文本的理解和处理能力。

**训练过程分析**

**数据集A**

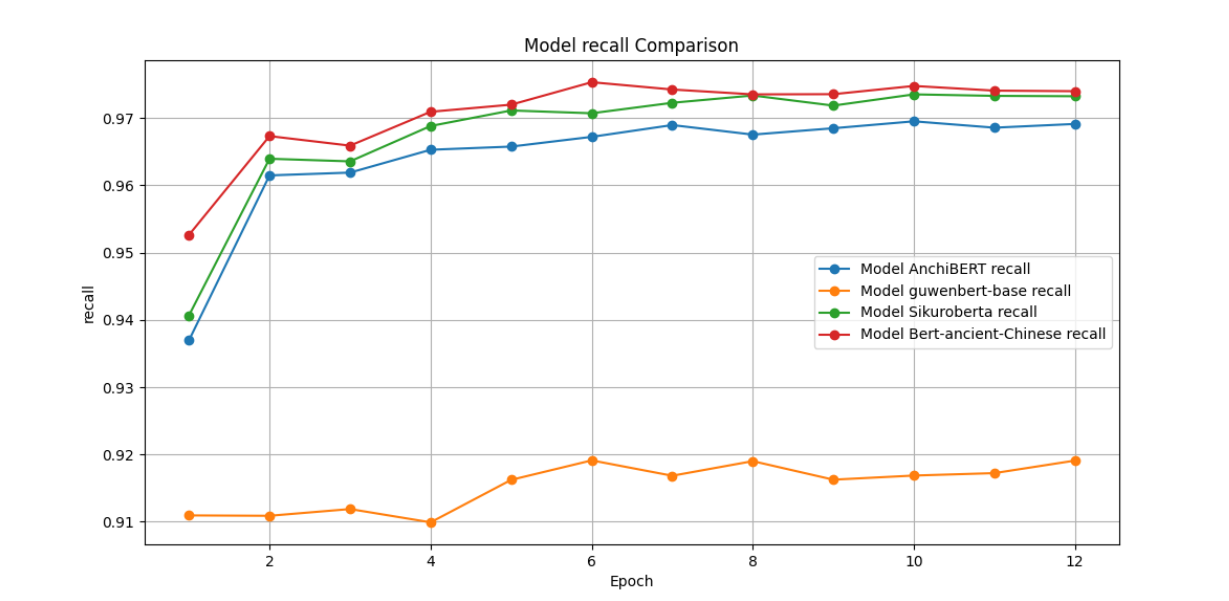
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Eval Precision** | **Eval Recall** | **Eval F1** | **Eval Accuracy** | **Eval Loss** | **Eval Runtime** |
| **2** | **0.889935668553033** | **0.9108930160142349** | **0.887439981938873** | **0.9108930160142349** | **0.3099377155303955** | **3.5996** |
| **4** | **0.8930562664726759** | **0.9099199288256228** | **0.8947124462689304** | **0.9099199288256228** | **0.30007246136665344** | **3.5699** |
| **6** | **0.9070822079888535** | **0.919122553380783** | **0.8958977944838243** | **0.919122553380783** | **0.2681475579738617** | **3.5973** |
| **8** | **0.9026081384861598** | **0.9190113434163701** | **0.9012913717862462** | **0.9190113434163701** | **0.27551373839378357** | **3.5942** |
| **10** | **0.8998557531360152** | **0.9168705516014235** | **0.9025673921036375** | **0.9168705516014235** | **0.27476179599761963** | **3.6135** |

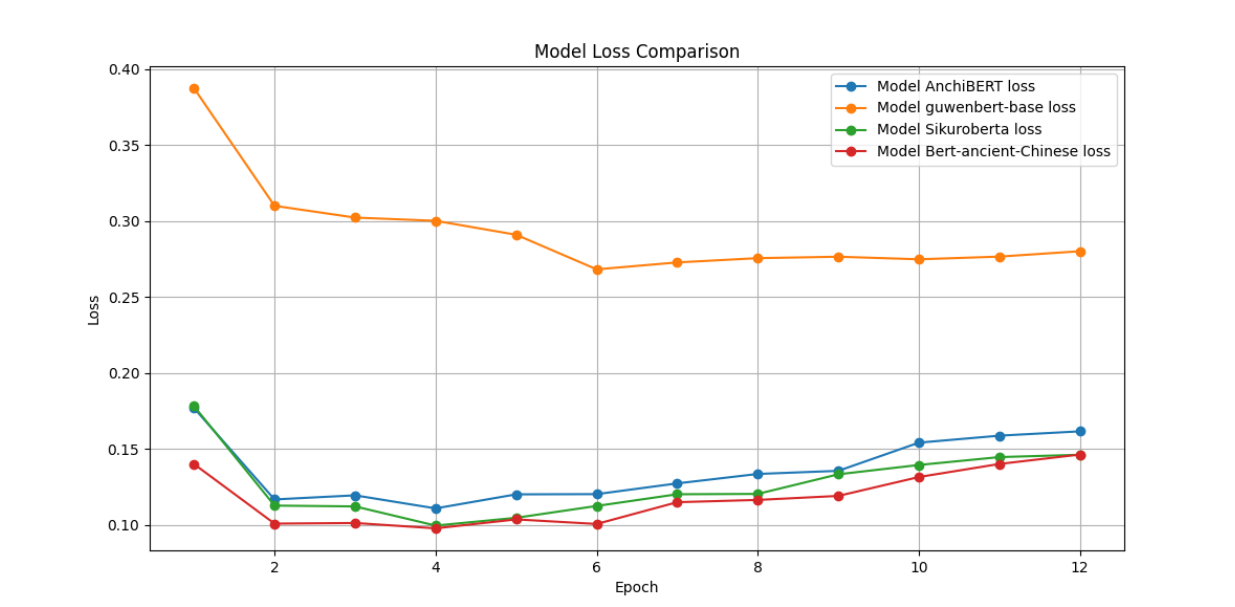
**数据集B**

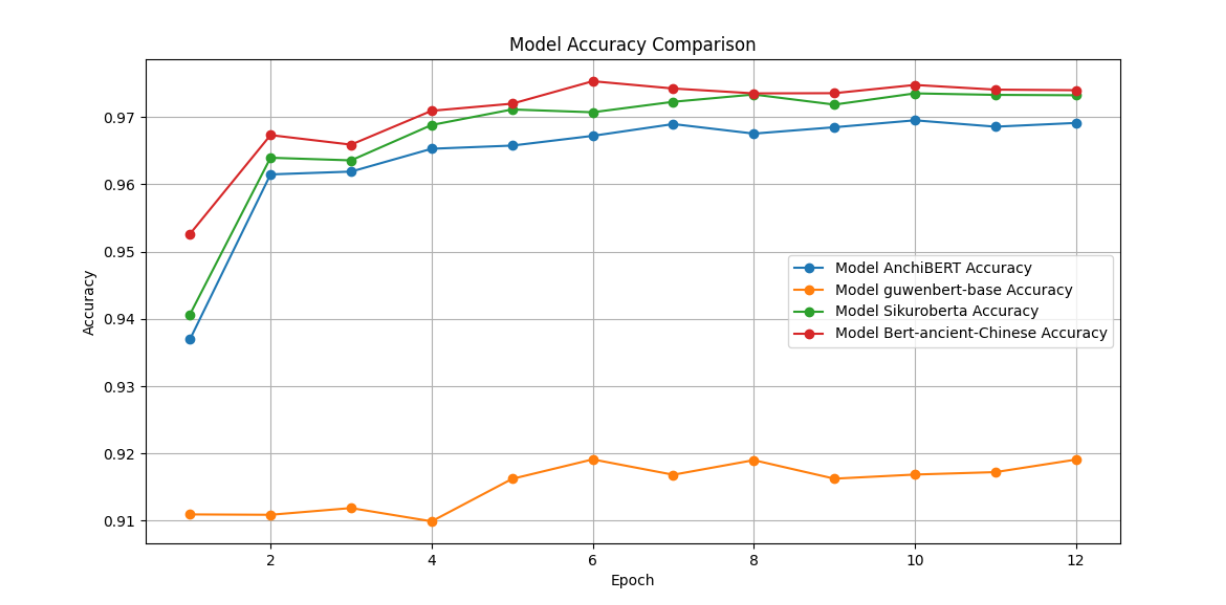
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Eval Precision** | **Eval Recall** | **Eval F1** | **Eval Accuracy** | **Eval Loss** | **Eval Runtime** |
| **2** | **0.9455200607648422** | **0.9664912840136054** | **0.9503247965618769** | **0.9664912840136054** | **0.14051271975040436** | **7.7432** |
| **4** | **0.9460385929088998** | **0.9663407029478458** | **0.9513311115749682** | **0.9663407029478458** | **0.1322239190340042** | **7.8783** |
| **6** | **0.9501682554545535** | **0.9662078373015873** | **0.9526954770627326** | **0.9662078373015873** | **0.1294545829296112** | **7.746** |
| **8** | **0.948499906885771** | **0.9653309240362812** | **0.953915135909169** | **0.9653309240362812** | **0.13299113512039185** | **7.5674** |
| **10** | **0.9488682087282262** | **0.9633645124716553** | **0.9544498117034105** | **0.9633645124716553** | **0.1366061121225357** | **7.5882** |

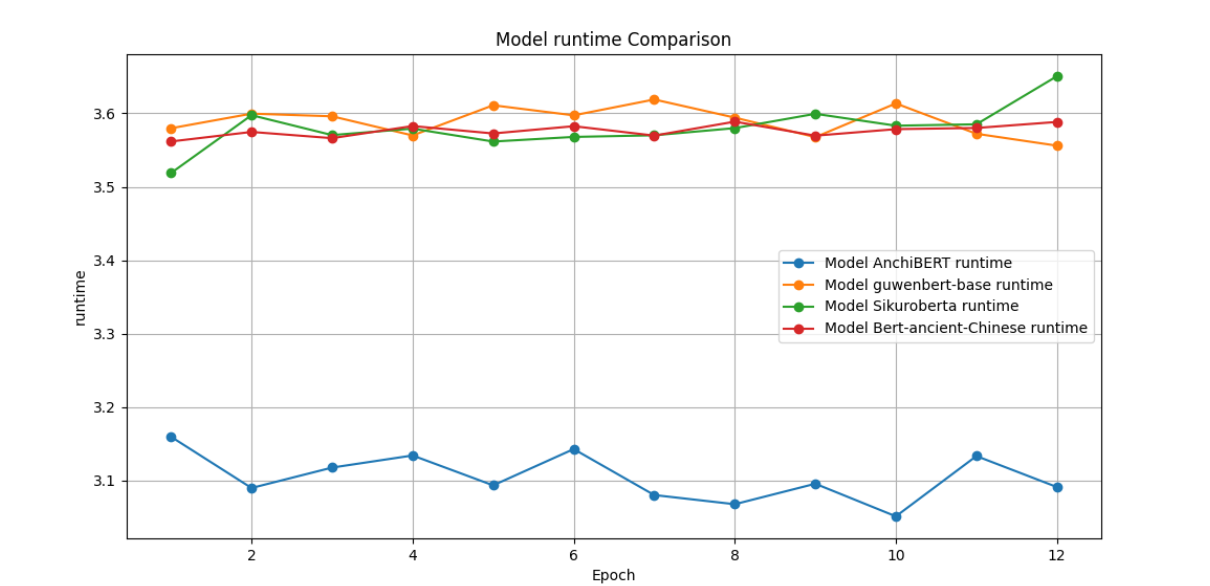
**比较分析**

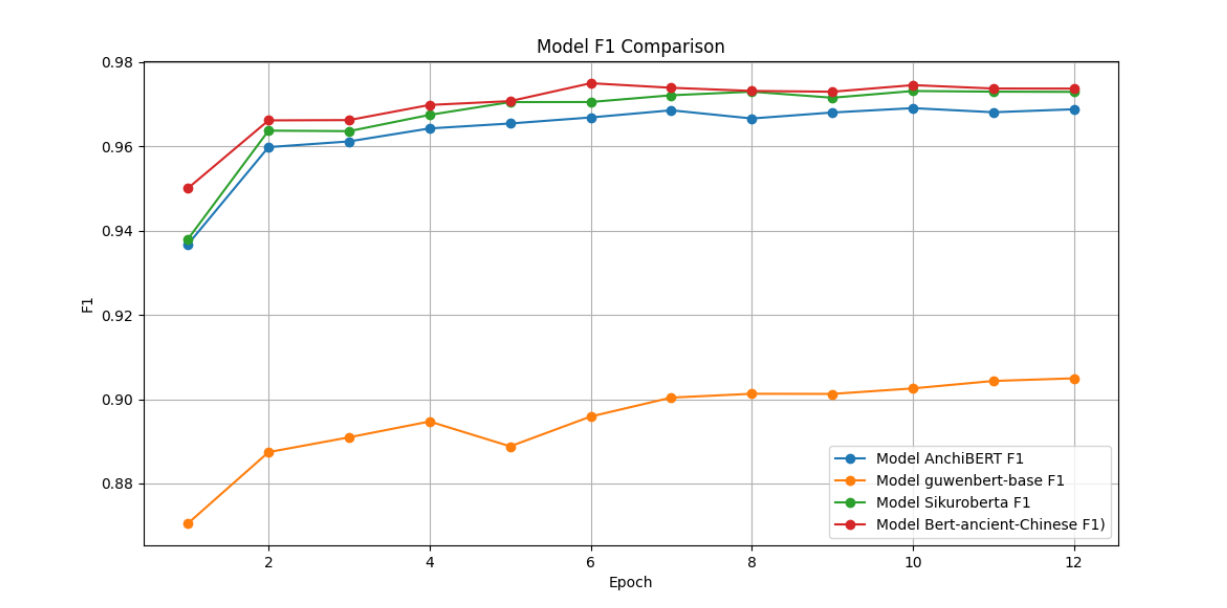
**数据集A**

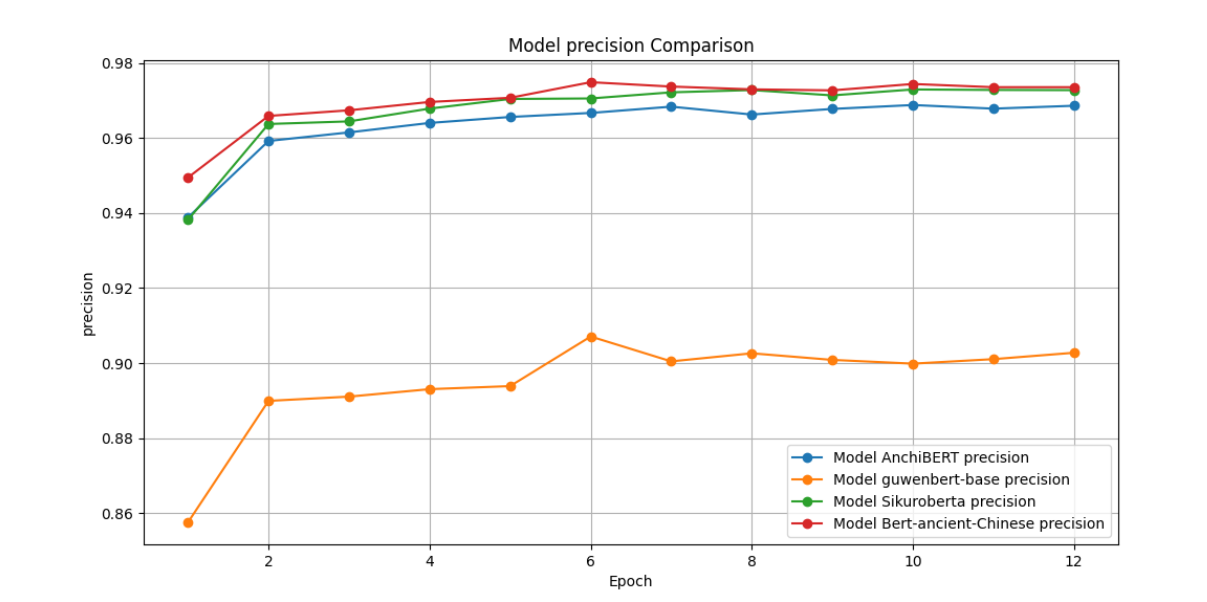












**guwenbert-base**

guwenbert-base在各方面表现较差。

**Bert-ancient-Chinese**

Bert-ancient-Chinese在整个训练中效果最好。

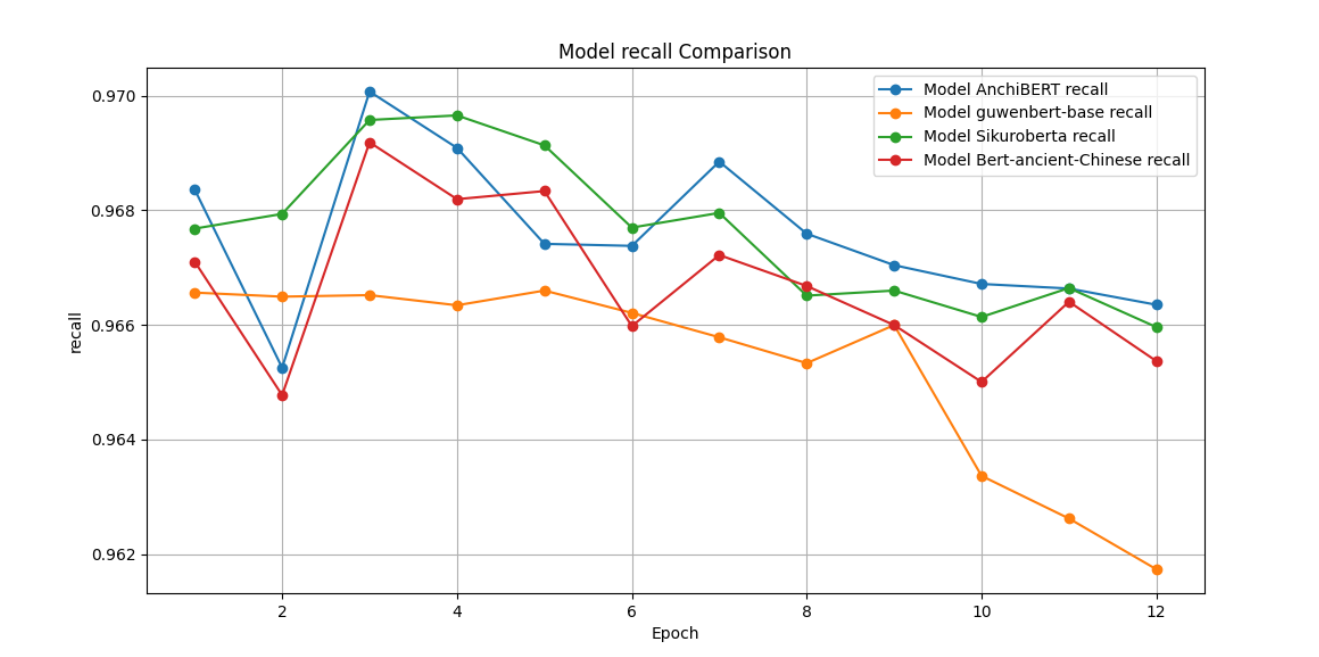
**AnchiBERT**

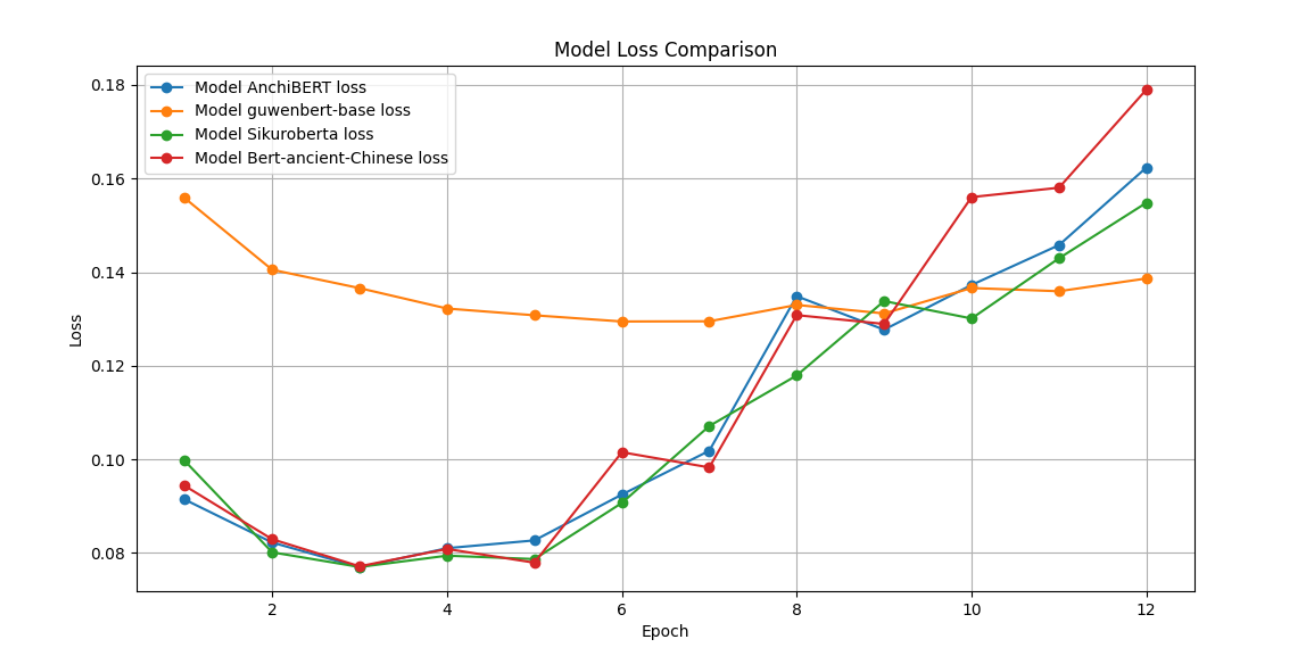
AnchiBERT在整个训练中不如SikuRoberta和Bert-ancient-Chinese，但是时间最短。

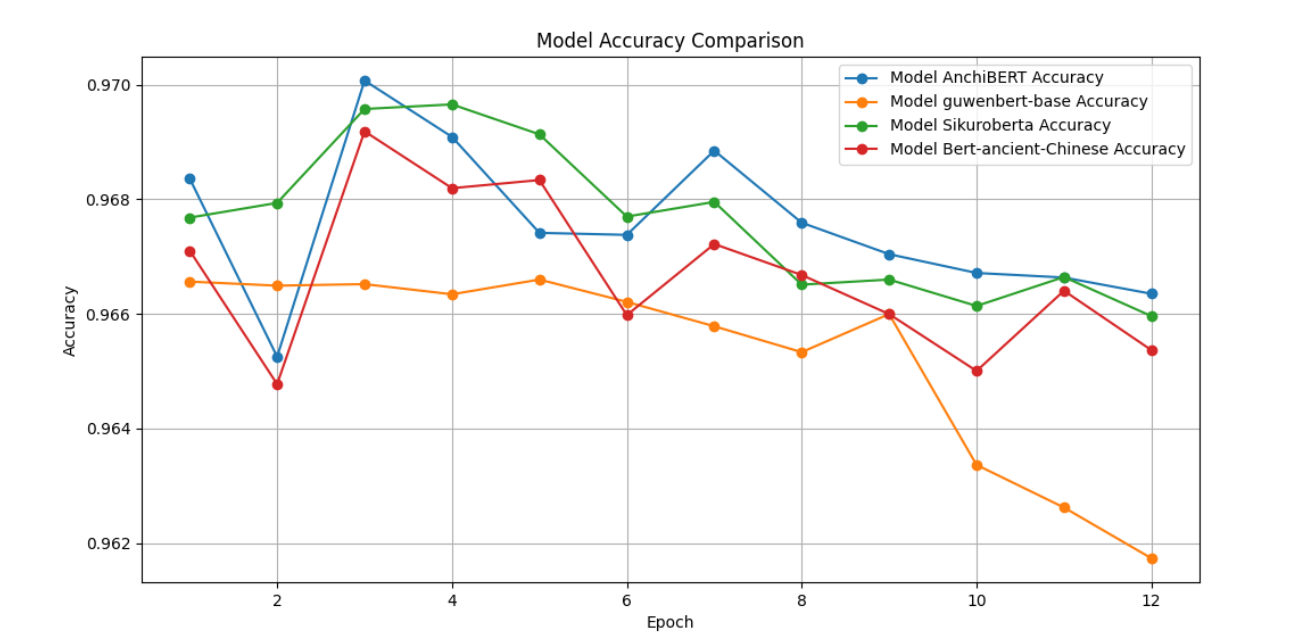
**SikuRoberta**

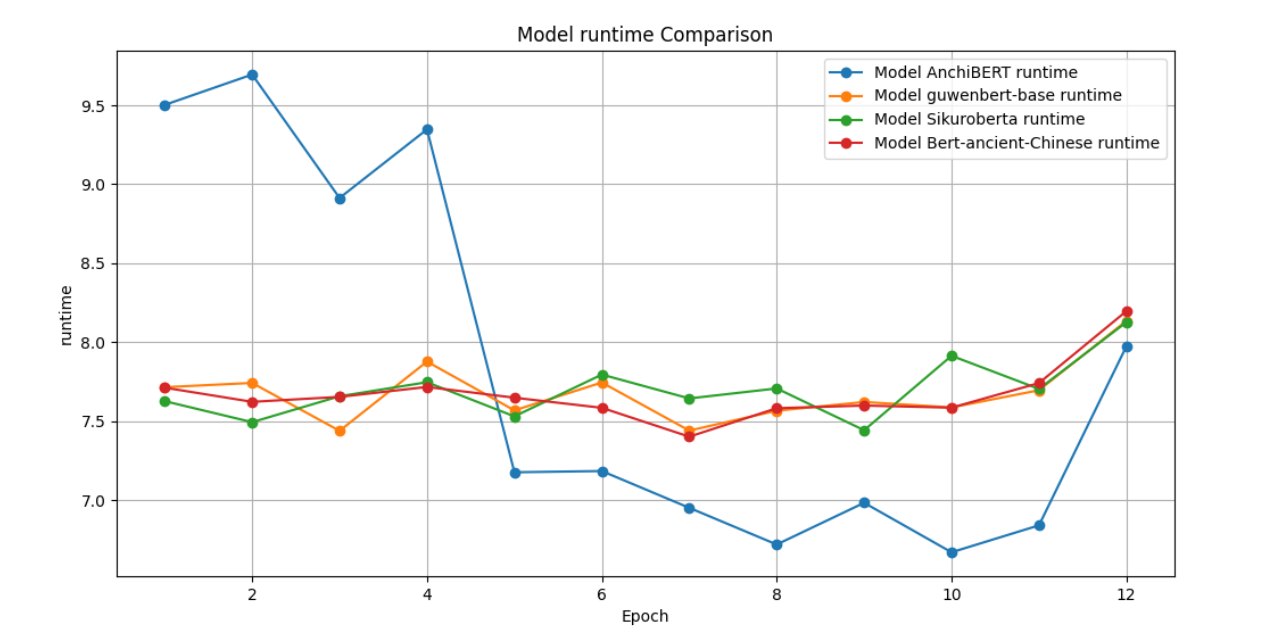
SikuRoberta各项指标稍逊于Bert-ancient-Chinese，训练效果相对优秀。

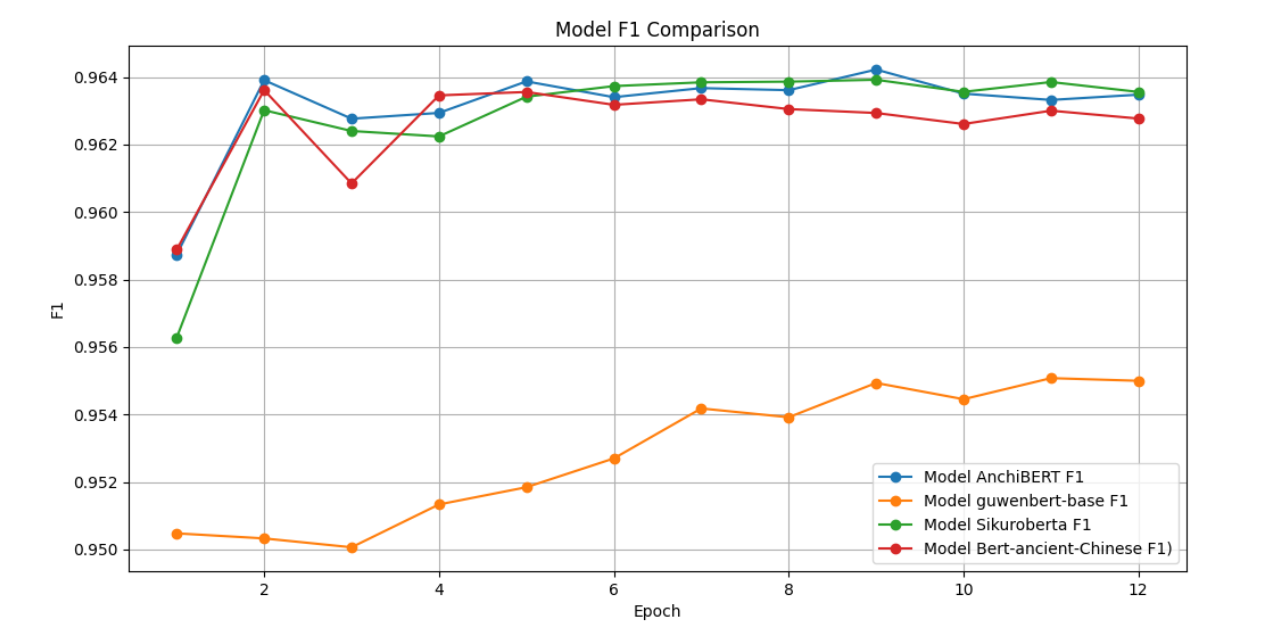
**数据集B**

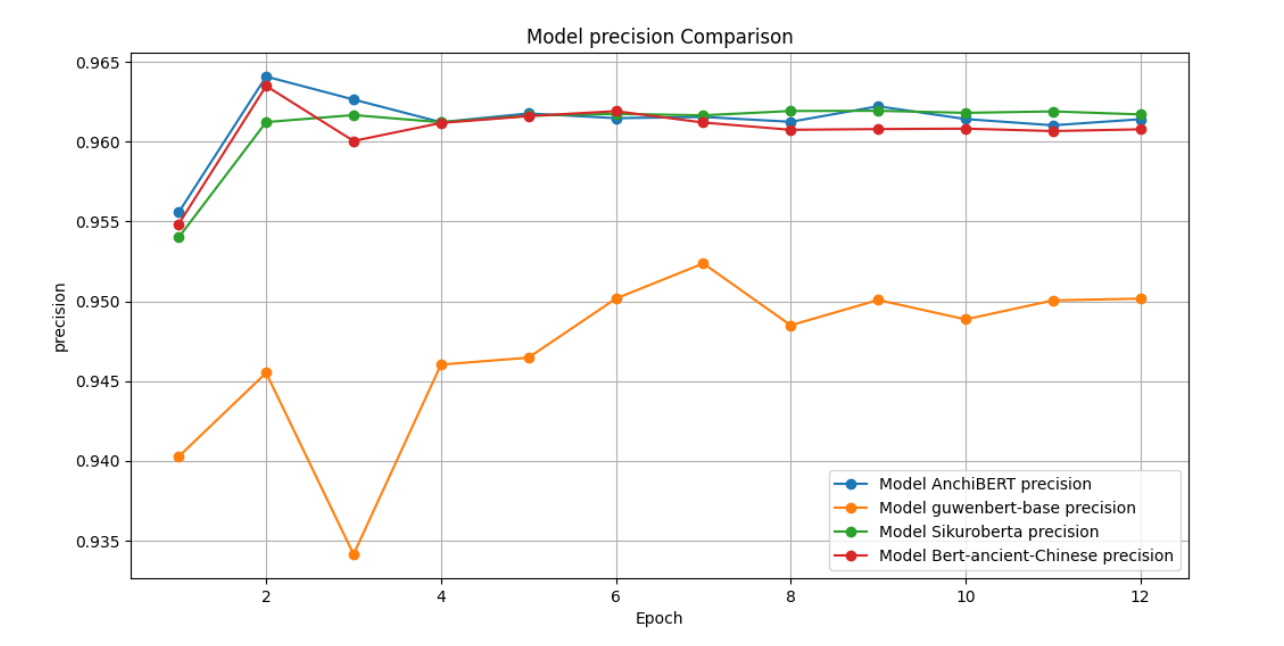












**guwenbert-base**

guwenbert-base在各方面表现不如其他三个模型。

**Bert-ancient-Chinese**

Bert-ancient-Chinese在precision和f1的表现与AnchiBERT和SikuRoberta类似，表现相对不错。

**AnchiBERT**

anchbert在其他方面表现不错，但在runtime中不太稳定。

**SikuRoberta**

SikuRoberta在precision和f1的表现与AnchiBERT和SikuRoberta类似，表现相对不错。

**综合比较**

Bert-ancient-Chinese和SikuRoberta表现都相对出色。

**实验整体分析**

**数据集A**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **模型** | **Micro-Average Precision** | **Micro-Average Recall** | **Micro-Average F1** | **Macro-Average Precision** | **Macro-Average Recall** | **Macro-Average F1** |
| **Bert-ancient-Chinese** |  |  |  |  |  |  |
| **SikuRoberta** |  |  |  |  |  |  |
| **AnchiBERT** |  |  |  |  |  |  |
| **guwenbert-base** |  |  |  |  |  |  |

**数据集B**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **模型** | **Micro-Average Precision** | **Micro-Average Recall** | **Micro-Average F1** | **Macro-Average Precision** | **Macro-Average Recall** | **Macro-Average F1** |
| **Bert-ancient-Chinese** |  |  |  |  |  |  |
| **SikuRoberta** |  |  |  |  |  |  |
| **AnchiBERT** |  |  |  |  |  |  |
| **guwenbert-base** |  |  |  |  |  |  |

**Micro-Average指标**反映了所有类别的总体性能，这意味着在考虑样本不平衡的情况下，guwenbert-base在整体上取得了最好的平衡性能。

**Macro-Average指标**展示了模型对所有类别同等对待的性能，guwenbert-base表现最佳，表明其对于少数类别的识别能力较强。