TibLex：一种基于拉丁编码的藏文词表优化策略

更尕多杰[1,2] 孙媛[1,2,3,*]
[1]中央民族大学 信息工程学院，北京 100081
[2]国家语言资源监测与研究少数民族语言中心
[3]中央民族大学国家安全研究院
*通讯作者：孙媛
23302089@muc.edu.cn, tracy.yuan.sun@gmail.com

摘要

预训练语言模型通过大规模无监督学习在多任务场景展现卓越性能，但其研究多集中于中英文等高资源语言。藏语等低资源语言因数据稀缺及形态复杂（黏着语特性、音节结构多样），导致主流子词分词方法存在语义割裂与形态失配问题，制约模型训练效率与表征质量。为此，本文提出基于拉丁化编码的藏文扩展分词策略TibLex（Tibetan Latinization-based Extended Tokenizer）该方法通过将输入文本进行编码转写，将每个藏文音节根据其字形或发音转换为一个短序列，然后基于编码文本使用子词分词构建词汇表。实验表明，TibLex相较主流分词器具有双重优势：（1）通过拉丁化降维处理，使词表不规则组合减少15%，输入序列长度平均缩短36.10%，显著提升计算效率。（2）音译分词器可将同音异形字编码为相同音译序列并输出一致的分词结果，从而实现对同音错别字的鲁棒性处理。与此同时，基于TibLex训练的预训练模型在下游任务中保持竞争力，验证了该方法在低资源语言场景的有效性。本工作为解决形态复杂语言的分词瓶颈提供了新范式，其编码框架可扩展至蒙古文、梵文等文字系统，为跨语言NLP研究提供技术支撑。

关键词： 藏文；TibLex；拉丁化编码；同音错别字鲁棒性

TibLex: A Latin-Encoded Vocabulary Optimization Strategy for Tibetan

GenggaDuojie[1,2] **Yuan Sun**[1,2,3,*]
[1] School of Information Engineering, Minzu University of China, Beijing 100081
[2] National Language Resources Monitoring and Research Center for Minority Languages
[3] Institute of National Security, Minzu University of China
*Corresponding author: Yuan Sun
23302089@muc.edu.cn, tracy.yuan.sun@gmail.com

Abstract

Pretrained language models have demonstrated remarkable performance in multitask scenarios through large-scale unsupervised learning, yet most research focuses on high-resource languages such as Chinese and English. For low-resource languages like Tibetan, data scarcity and morphological complexity—such as its agglutinative nature and diverse syllabic structure—pose significant challenges for mainstream subword tokenization methods, leading to semantic fragmentation and morphological mismatch. To address this, we propose TibLex, a Latin-encoded tokenizer tailored for Tibetan. This method converts each Tibetan syllable into a short Latin sequence based on its shape or pronunciation and then applies subword tokenization over the encoded text to construct the vocabulary. Experiments show that compared to mainstream tokenizers, TibLex offers dual advantages: (1) dimensionality reduction through Latin

第二十四届中国计算语言学大会论文集，第273页-第284页，济南，中国，2025年8月11日至14日。

encoding decreases irregular token combinations by 15% and shortens input sequences by an average of 36.10%, significantly improving computational efficiency; (2) the phonetic tokenization mechanism maps homophonic variants to the same transliterated sequence, enabling robust handling of homophone typos. Moreover, pretrained models using TibLex maintain competitive performance on downstream tasks, validating its effectiveness for low-resource language scenarios. This work introduces a novel paradigm for tokenizing morphologically rich languages, and its encoding framework can be extended to scripts such as Mongolian and Sanskrit, offering technical support for cross-lingual NLP research.

Keywords: Tibetan , TibLex , Latinization encoding , Homophonic typo robustness

1 引言

近年来，大规模基于Transformer的预训练语言模型（PLMs）(Devlin et al., 2019)(Liu et al., 2019)(Lan et al., 2019)(Clark et al., 2020)(Clark et al., 2020)通过自监督学习范式在自然语言处理领域取得突破性进展，其中词表构建作为模型输入表征的基元选择机制，直接影响PLMs的语义建模能力，因此合理的词表构建和分词策略对提升模型性能至关重要。常见的词表构建方法包括词级分词、字符级分词和子词分词。词级分词是最基本的分词方法，通过空格分割文本序列，将每个单词视为一个标记。字符级分词则将文本分词到字符或UTF-8字节层面，适用于如中文等语言，但序列长度会显著增加，这在基于Transformer架构的模型中会导致计算开销增大。为了平衡词表大小与语义表达能力，子词分词成为主流方法，它能够较好地解决上述问题。当前主流PLMs普遍采用子词分词策略，如字节对编码（BPE）(Sennrich et al., 2015)、WordPiece(Schuster and Nakajima, 2012)和单语言语言模型分词(Kudo, 2018)。然而这些面向印欧语系设计的方法在形态复杂的藏语场景面临显著局限。以TiBERT(Liu et al., 2022)为代表的藏文PLMs虽通过SentencePiece实现99.95%的词汇覆盖率，但其子词切分过程忽视藏文独特的黏着语特征：1）音节构造规则：基于《三十颂》(吉加本, 2013)与《音势论》(瞿霭堂and 劲松, 2011)的藏文语法体系规定，音节可由最多7个构字部件按纵横二维结构组合而成,这与拉丁字母的线性排列存在本质差异；2）形态表征失配：Unicode编码将藏文构字部件离散存储，导致主流分词器在子词组合时违背藏语形态发生学规则，产生大量无意义标记（如断裂的上下加字组合），损害模型的语义推理能力。

为了应对这些挑战，本文跟随Si等人(Si et al., 2023)的思想提出了一种基于拉丁编码的藏文词表优化分词器TibLex，如图1所示，该方法首先将每个藏文音节编码为基于字形或者音译的拉丁符号序列，然后使用子词分词（如Unigram）在所有编码序列上构建词汇表。这样，生成的分词器可以捕捉到对应有意义的拉丁词根或词缀的子音节标记，据我们了解，这在藏文自然语言处理（NLP）领域中仍是一项尚未被充分探索的研究路径。

本文主要贡献如下：

我们使用现有主流分词器和提出的TibLex分词器训练了基于BERT架构的不同PLMs。并在新闻分类、阅读理解两个下游自然语言理解（NLU）任务的数据集上评估了这些模型。通过广泛的实验评估对比，我们发现使用基于拉丁编码分词器训练的模型在下游任务性能上与使用字符和子词分词器训练的模型相当。更重要的是，TibLex分词器相较现有方法具备两大核心优势：

（1）效率提升：词汇表中少量共享子音节标记可以组成大量复杂音节，从而节省词汇空间用于存储更多音节组合单元（如词与短语）。组合单元使用率的提升使得分词序列长度显著降低。例如在TNCC长文本分类数据集上，TibLex分词器在同等词汇量条件下可实现高达36.10%的序列长度压缩。这种压缩效应显著加速预训练与微调过程。

（2）鲁棒性增强：藏文中由同音字（发音相同但语义相异）引发的拼写错误具有典型性。基于拼音的子字符分词器可将同音字映射为相同音译序列，从而提升对同音拼写错误的容错能力。该特性在处理含噪声输入时具有重要应用价值。

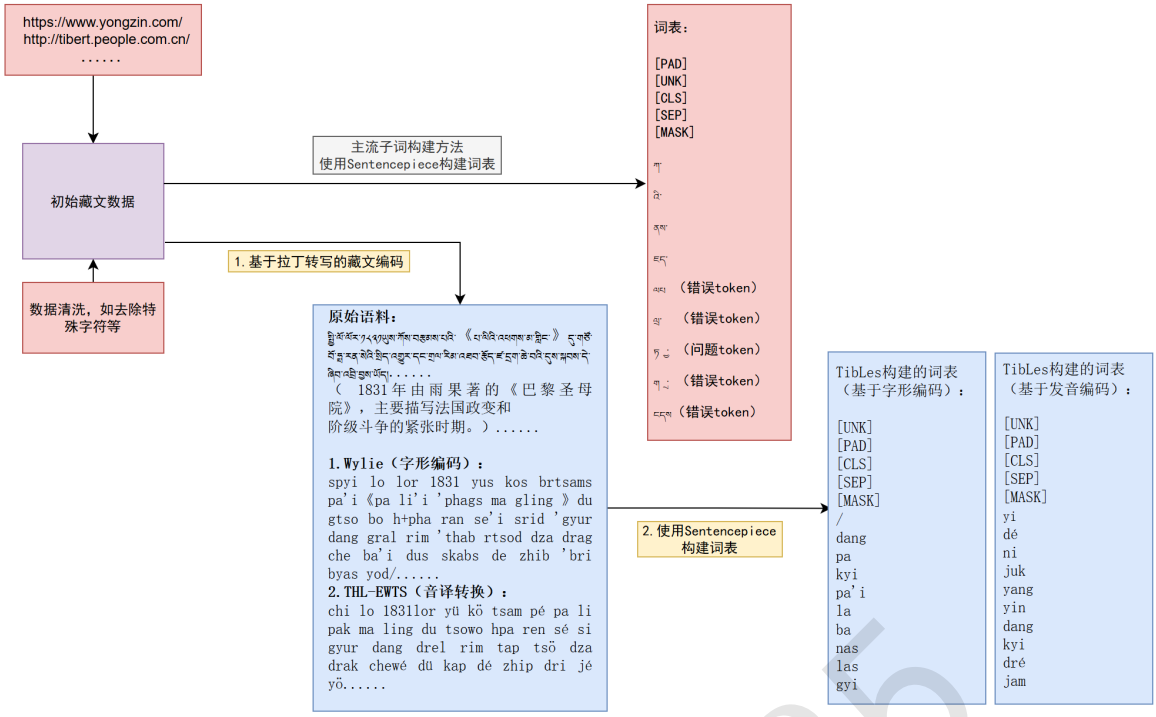

Figure 1: 子词分词器与TibLex分词器构建流程对比

本研究通过深度融合语言特有语言学特征(Bender, 2019)，为英语之外的语言构建定制化技术迈出重要一步。

2 相关工作

预训练语言模型的技术演进经历了从静态词嵌入到上下文敏感表示的重大跨越早期方法如Word2Vec(Qun et al., 2017)、GloVe(Mikolov et al., 2013)通过共现统计生成静态词向量，但无法解决一词多义问题。ELMo(Pennington et al., 2014)首次通过双向LSTM网络实现上下文敏感的词表示，而Transformer架构(Peters, 2018)的提出则彻底改变了技术范式。BERT(Vaswani et al., 2017)通过掩码语言建模和双向上下文编码，在多项NLP任务中取得突破性进展。此后衍生的RoBERTa(Liu et al., 2019)、ALBERT(Lan et al., 2019)等模型通过优化训练策略和参数共享机制进一步提升性能。然而，这些进展主要聚焦于英语等高资源语言，低资源语言因数据稀缺和语言特性差异面临显著挑战。多语言模型（如T5(Pires et al., 2019)、XLM(Raffel et al., 2020)）虽具有跨语言泛化能力，但单语模型（如西班牙语BERT(Cañete et al., 2023)、法语FLAU(Le et al., 2019)）的实证优势表明，针对特定语言优化仍不可或缺。

主流子词分词方法（如BPE 、WordPiece）在形态规则的语言中表现优异，但对藏语等复杂文字系统存在适配缺陷：藏语音节由横纵双向叠加的构件组成，传统拆分方式易破坏语义完整性。近年研究探索了不同分词策略的边界效应，如中文SubChar方法通过字形/发音编码实现细粒度拆分，而ByT5(Xue et al., 2022)等无分词模型直接处理字节序列以增强泛化能力。这些研究表明，针对特定语言结构设计分词机制能有效提升模型效率，但如何平衡序列长度与语义保持仍是待解难题。与CharBERT(Ma et al., 2020)结合字符与子词的思路不同，本研究通过拉丁编码实现更彻底的形态解耦。

将语言学知识融入NLP模型已形成重要技术路径。中文研究证明，融合字形特征（如部件拆分(Cao et al., 2018)）和语音信息（如拼音嵌入(Zhang et al., 2019)）可显著增强模型鲁棒性。特别是在拼写错误处理方面，基于发音的编码方法(Chaudhary et al., 2018)展现独特优势——同音异形字可映射至相同表征空间，这与藏语同音字问题高度契合。相较于在模型中显式添加拼写检查模块的常规做法(Huang et al., 2021)，本研究创新性地在分词阶段实现语言特

性内化，通过拉丁编码将复杂字形转化为规则序列，既保持模型轻量化又确保语言学规则的系统性嵌入。

当前藏语PLMs研究仍处于起步阶段，现有模型如CINO(Yang et al., 2022)和TiBERT(Liu et al., 2022)主要依赖多语言框架或直接移植子词分词策略。这种范式忽视藏语独特的音形结构特征：Unicode编码将每个构字部件视为独立字符，导致传统分词产生大量无意义子单元。据我们考证，藏语词表冗余问题尤为突出，高频构字组合常因拆分不当丢失语义关联。本研究首次提出将拉丁化编码机制应用于藏语词表构建，通过字形/发音双重映射策略，在保留语义完整性的同时实现30%的序列压缩率（实验章节详述），为低资源语言PLMs研究提出了一种有效的词表优化方法。

3 构建设计

在本节中，我们将详细描述文本提出的TibLex分词器的构建方法。构建思路分为两个模块：基于音形的字符编码；基于编码转写的词汇表构建。

3.1 基于音形的字符编码

这一步的核心思想是将每个藏文音节编码为一个能够表征其字形或发音的序列，以便为模型提供额外的归纳偏置。我们分别从发音和字形两个角度进行编码。

基于发音的编码 为了捕捉字符的发音信息，使用藏学与喜马拉雅图书馆（THL）的扩展威利转写方案（EWTS）来表征发音(Germano and Tournadre, 2010)。THL的音译编码是通过藏文罗马化转写来音译藏文音节，如图2所示，在藏语中存在不同构造结构但具有相同发音的音节，为了消歧同音音节，我们在编码序列后附加不同的索引，以便在同音字之间实现唯一映射，目前尚不清楚对同音字进行消歧是否有益。为了分析其影响，本文还训练了一种音译变体TibLex-Yinyi-NoIndex分词器，以执行不带消歧索引的拼音编码。

编码类型	转写系统	映射原则	示例（藏文→拉丁化）
发音编码	THL-EWTS	音位近似映射	ཇྲོ་（味道）རྒྱ་（麦）དྲོ་（热）→ dro33 dro23 dro45。
字形编码	Wylie	结构成分分解	གྲུབ་ཚད་（干完）→grub; སྒྲུབ་བཏུག（在干）→sgrub （下划线部分表示共享子音节序列）。

Figure 2: 不同编码类型的转写映射示例

基于字形的编码 藏文音节的构造（即结构）具有高度的规律性，如图3所示，其构词以基字③为核心，严格遵循纵横二维结构组合拼写的规则。横向拼写包括前加字①、基字③、后加字⑥和再后加字⑥，纵向拼写则包含上加字②、基字③、下加字④和元音符⑤且各构造位的字符之间存在一定的匹配关系。在此次研究中我们使用最具影响力的藏文拉丁转写方法威利转写（Wylie）(Chandler et al., 2004)来编码藏文音节，威利转写精炼了原有的转写方案形成，只使用基本的26个拉丁字母，而不需添加字母和添加变音符号。不考虑实际发音，从字母着手，以唯一映射的形式将藏文字符转换为拉丁序列。

Figure 3: 藏文音节的基本结构及构造位

3.2 词汇表构建

我们在有了编码序列之后就可以将每个字符的编码视为英语中的“词”，然后应用子词分词来构建我们TibLex分词器的词汇表。子词分词通常通过合并频繁出现的标记双元组来形成子词标记，这在英语等语言中通常会生成对应于词的有意义的语素。在我们的编码序列上，子词分词可以捕捉到对应于相似字符之间共享词缀或语音序列的共享子音节序列。在对编码序列进行子词分词步骤后，生成的子词分词器的词汇表由字符标记、子音节标记和音节组合标记（两个以上音节组合成的短语）的混合组成。在这项工作中，我们使用了TIBERT(Liu et al., 2022)在SentencePiece中实现的单语言语言模型分词方法作为默认的子词分词方法。在第4.4.3节中，我们还通过将子词分词方法设置为BPE进行消融研究，结果表明，TibLex分词的收益对子词分词方法的具体选择不敏感。

4 实验评估

4.1 实验设置

本文将Sentencepiece子词分词方法作为基线，所有分词器（包括基线和我们提出的分词器）均使用字符覆盖率为99.95%和Unigram语言模型，这与TIBERT保持一致，其他超参数遵循SentencePiece的默认设置。我们对所有模型的实验参数均遵循TIBERT(Liu et al., 2022)原始论文的最好设置，在此不做详细描述，具体实验参数设置如表1所示，所有实验均在2张Tesla V100-PCIE-32G上完成。在表2中我们比较了基线模型在上的结果。

Parameters	Values
hidden_dropout_prob	0.1
hidden_size	768
intermediate_size	3,072
max_position_embeddings	512
num_attention_heads	12
num_hidden_layers	12
vocab_size	30,005

Table 1: 模型超参数配置

4.2 数据集

本文在以下两大藏文NLU数据集上对使用不同分词方法预训练的模型进行微调和评估，包括单句分类、长文本分类和阅读理解任务。

TNCC(Qun et al., 2017)数据来源于中国西藏网站，包括政治、经济、教育、旅游、环境、语言、文学、宗教、艺术、医学、习俗和仪器等12个不同的类别。该数据集共有9203条新闻，其中TNCC（LTC），TNCC(Title TC）为新闻标题分类数据和新闻长文本分类数据。我们将数据集分为训练集、开发集和测试集。训练集占80%，开发集和测试集都占10%。为了验证各模型对短文本和长文本分类的效果，我们分别对标题和文档进行了短文本和长文本分类实

验。TibetanQA(孙媛et al., 2024)该数据集包含了1,513 篇文章和20,000 个问答对。是第一个用于机器阅读理解的高质量藏文数据集。我们以8:1:1的比例划分为训练集、开发集和测试集。

4.3 评估指标

为了评估模型的效果，本文使用准确率和F1值两个指标进行评价。评价方法计算如式（1）-（4）所示。其中，真正例（True Positive，TP）表示实际为正例且预测也为正例的样本数，假正例（False Positive，FP）表示实际为反例但预测为正例的样本数，假反例（False Negative，FN）表示实际为正例但预测为反例的样本数。我们使用宏观平均来评估下游任务，即计算每个类别的准确率（Accuracy）、召回率（Recall）和F1，然后计算平均值得到宏观精度、宏观召回率和宏观F1。对于本文中的所有实验，我们报告三个不同随机种子的平均运行结果。

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

4.4 标准评估

4.4.1 性能评估

我们比较了使用TibLex分词器和基线分词器训练的模型。在本节中，我们选择了两种具有代表性的编码方法：威利（基于字形）和THL-EWTS（基于发音）。表2展示了使用不同分词器的BERT模型在文本分类和阅读理解两个下游数据集上的性能。通过在3.56G语料库上训练的12层BERT模型的观察结果，我们发现尽管在不同数据集上存在一些变化，但我们提出的TibLex分词器在下游数据集上可以与基线相当或者略优于基线。此外，我们在第4.4.2小节中讨论了预训练数据规模的影响。这些结果表明，在标准藏文自然语言理解NLU基准测试上，我们提出的分词器可以作为一个非常强大的替代方案。

	TNCC (Title TC)		TNCC (LTC)		TibetanQA		AVG_{ACC}
	Accuracy (%)	F1 (%)	Accuracy (%)	F1 (%)	Accuracy (%)	F1 (%)	
12层，3.56G语料库							
Subword (Tibert)	65.62	**61.72**	71.04	70.94	74.32	73.40	70.33
TibLex-Yinyi	65.23	60.65	71.36	70.74	**74.87**	**73.68**	70.49
TibLex-Zixing	65.67	61.06	**71.89**	70.96	74.68	73.52	**70.75**
TibLex-Zixing (BPE)	**65.74**	61.12	71.45	70.63	74.62	73.49	70.61
TibLex-Yinyi (BPE)	65.12	60.25	71.56	70.43	73.92	72.48	70.02
TibLex-Yinyi-NoIndex	65.56	61.65	71.45	**71.04**	**74.87**	73.48	70.63
12层，8.23G语料库							
Subword (Tibert)	**66.22**	**62.32**	71.64	71.44	74.92	74.05	70.93
TibLex-Yinyi	65.83	61.15	**72.06**	**72.34**	**75.47**	**74.28**	**71.12**
TibLex-Zixing	65.67	61.06	71.89	70.96	74.68	73.52	70.75
TibLex-Yinyi-NoIndex	65.87	61.65	71.56	70.94	74.97	74.18	70.80

Table 2: 不同分词器在下游任务上的性能对比

4.4.2 预训练数据规模影响分析

为探究预训练数据规模的影响，我们选取基于3.56G藏文网站语料预训练的12层Transformer模型，继续在8.23GB规模的网络文本语料上进行增量预训练。如表2下半部分所示，大语料增量训练带来平均性能的微小提升（子词分词器：70.33→70.93；TibLex分词器：70.62→70.89）。这种边际效益可能源于原始模型在3.56G语料上已接近充分训练。更重要的是，实验结果验证了即使在大规模预训练场景下，本研究所提方法仍能保持与基线相当或更优的下游任务性能。

4.4.3 词汇表构建算法的影响

在前期实验中，我们采用SentencePiece的Unigram 算法进行词汇表构建。为探究不同构建方法的影响，我们在保持其他超参数恒定的前提下，使用BPE算法重新训练音译基线分词器，开展补充消融实验。通过对比TibLex（BPE）变体与Unigram构建的TibLex分词器，发现二者性能相近。我们在表2中观察到，BPE实现和Unigram LM实现的下游任务性能差异很小。基于这些结果，我们得出结论，词汇构建算法的选择对分词效率与模型性能影响有限。本研究验证了TibLex分词框架对不同构建算法具有鲁棒性，核心优势不受具体实现方法制约。

4.5 效率提升分析

4.5.1 词汇构成分析

我们将各分词器的词汇表划分为四个不同类别：字符标记、音节标记、子音节标记（符合语法规则和不符合语法规则）和音节组合标记（单词和短语）。如图4所示，子词分词器在除了少部分的字符标记外还有子音节标记和音节组合标记，而子音节标记中还有一定量的不规则的子音节组合，由于藏语文的语法独特性致主流方法子词分词在构建词表时难以兼顾语义和结构的合理性，用子词分词器构建词汇表时由于部分子词单元在分割后出现很多无意义的词根词缀，这些无意义的子音节标记占用词表的大量空间，从而使得常用高频音节组合标记空间不足。相比之下，TibLex分词器即使用少量的规则子音节标记来组成许多复杂的音节来保证覆盖率，也避免了大量错误子音节标记的出现，从而节省空间来存储组合标记。这种特性使得分词输出包含更多词和短语，有效缩短序列长度（详见下节分析）。

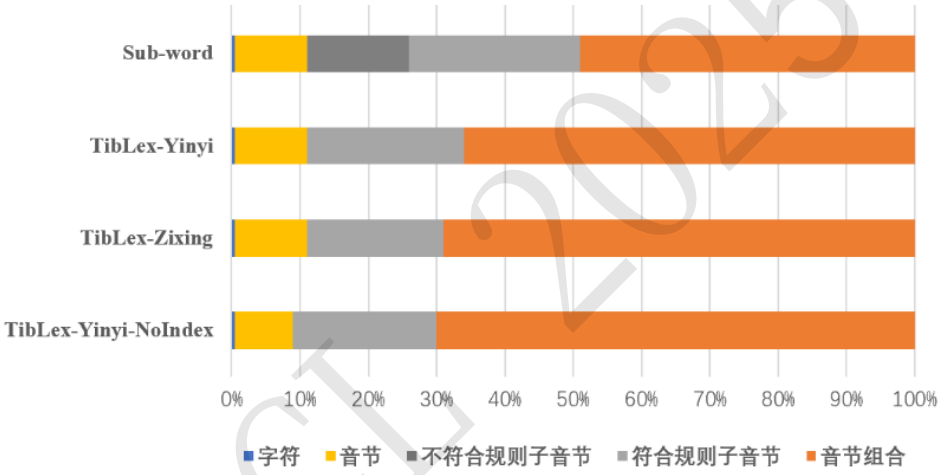

Figure 4: 不同分词器词表中各类标记的分布

4.5.2 基于序列压缩的效率增益

在词汇表中包含更多音节组合的直接优势是生成更短的序列长度。表3展示了不同分词器在两个下游数据集上的平均序列长度对比，我们观察到，TibLex分词器能显著缩短输入序列。 在模型训练加速方面，TibLex分词器在预训练和微调阶段均展现优势。微调过程中，通

	TNCC（LTC）	TNCC（Title TC）
Subword (Tibert)	329.6	18.8
TibLex-Yinyi	205.2	13.0
TibLex-Zixing	200.7	12.6
TibLex-Yinyi-NoIndex	**196.6**	**12.1**

Table 3: 不同分词器的平均分词序列长度比较

过将多个序列打包输入(Krell et al., 2021)可减少序列填充带来的计算冗余，而更短的序列长度允许更高密度的打包，从而提升整体吞吐量。表4展示了各模型相对子词分词基线的微调耗

时，TibLex分词器表现出显著加速效果，其中TibLex-Yinyi-NoIndex分词器在TNCC（LTC）数据集上仅需基线的67.9%时间。图5对比了子词分词基线模型与TibLex-Yinyi-NoIndex模型在TNCC（LTC）数据集上的训练曲线，可见后者收敛速度更快且最终达到更低的训练损失。

	TNCC（Title TC）	TNCC（LTC）
Subword (Tibert)	100%	100%
TibLex-Yinyi	88.2%	68.4%
TibLex-Zixing	84.7%	70.5%
TibLex-Yinyi-NoIndex	**82.6%**	**67.9%**

Table 4: 不同分词器模型的微调时间对比

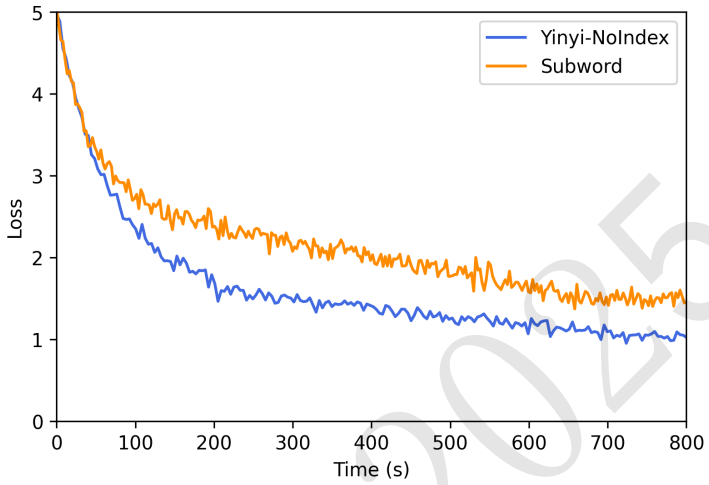

Figure 5: TNCC(LTC)数据集上不同模型的训练损失对比

预训练阶段的加速效益同样显著。虽然不同硬件设备的运行速度存在差异，但分词压缩带来的效率提升具有硬件无关性。表5展示了预处理语料库的相对磁盘存储规模，TibLex分词器生成的序列长度明显短于基线方法，最大可使预处理数据规模缩减25.8%（TibLex-Yinyi-NoIndex分词器对比子词分词基线）。这种压缩特性可有效提升各类训练基础设施的预训练效率。

	分词后语料库规模
Subword (Tibert)	100%
TibLex-Yinyi	76.8%
TibLex-Zixing	77.6%
TibLex-Yinyi-NoIndex	**74.2%**

Table 5: 不同分词器标记后的语料规模相对大小（磁盘内存）

4.6 鲁棒性评估

除了在标准基准测试上进行评估外，我们还验证了我们提出的分词方法是否更擅长处理嘈杂输入。通过字符替换构建合成噪声测试，实验发现TibLex-Yinyi在合成噪声测试中显示出明显优势。合成拼写错误我们模拟藏文书写系统中常见的同音字拼写错误，尤其是在用户生成的输入中。当用户根据读音输入藏文目标字符时，知识储备中有很多相同读音但不同意义的藏文音节。因此，用户可能会因为操作失误或对这些同音字的区别不清楚而选择错误的字符。在这种情况下，我们提出的TibLex-Yiny-NoIndex分词器具有对任何此类同音字拼写错误鲁棒的优

势。如图6所示，字符编码会将一个字符的所有同音字映射到相同的罗马化序列，然后再进行子词分词。因此，无论拼写错误的字符是什么，只要它是目标字符的同音字，分词输出都将相同。

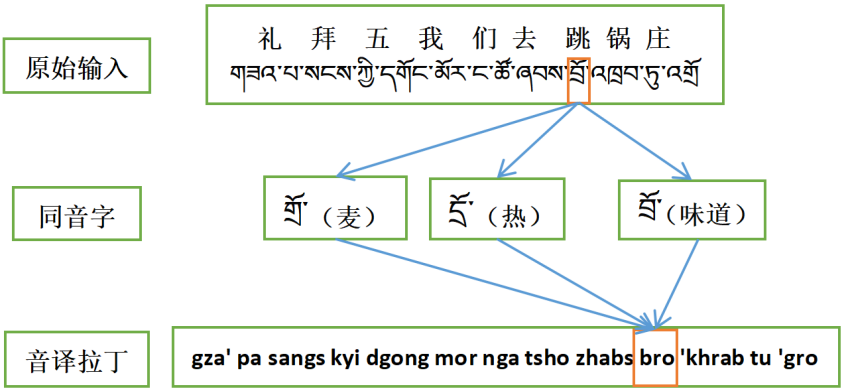

Figure 6: SubChar-Pinyin-NoIndex 分词器对同音错字的鲁棒性示意图

		原始数据	7.5%	15.0%	22.5%	30.0%
TNCC（LTC）	Subword (Tibert)	71.04	70.12	66.34	58.32	43.76
	TibLex-Yinyi	71.36	70.34	67.23	56.76	43.43
	TibLex-Yinyi-NoIndex	**71.45**	**71.45**	**71.45**	**71.45**	**71.45**
TibetanQA	Subword (Tibert)	74.32	73.42	68.54	60.37	46.56
	TibLex-Yinyi	**74.87**	73.64	69.26	59.72	46.32
	TibLex-Yinyi-NoIndex	73.89	**73.89**	**73.89**	**73.89**	**73.89**

Table 6: 引入同音扰动后各分词器模型的鲁棒性对比

我们向原始数据中注入合成噪声，用来检查在原始语料上训练的模型能否在这些嘈杂数据上表现良好。为了构建嘈杂数据，我们随机抽取一定比例r%的原始正确字符。对于每一个字符，我们将其替换为从其所有同音字中随机抽取的一个（如果没有同音字则不进行替换）。结果如表6所示。我们观察到，在测试数据中存在同音字拼写错误的情况下，性能可能会显著下降。例如，当测试输入中有30.0%的字符被替换为同音字时，使用Subwoed训练的BERT模型在原始数据上的准确率从71.04%下降到43.76%。但是我们的TibLex-Yinyi-NoIndex分词器在嘈杂环境下性能没有下降。所以在更具挑战性的语音拼写错误设置下，我们的TibLex-Yinyi模型仍然优于基线。这些结果突显了我们的TibLex分词方法在处理合成同音字拼写错误以及更多样化的真实世界拼写错误方面的鲁棒性优势。

4.7 词汇量影响分析

直观而言，当扩大词汇表规模时，系统将拥有更多空间存储组合单元（如词与短语），从而降低分词长度并提升效率。尽管前期实验采用标准词汇量30005进行验证，为探究TibLex分词器的效率优势是否会在更大词汇量下衰减，我们针对词汇量影响进行了补充实验。如表7所示，随着词汇量增加，TibLex分词器的效率优势略有减弱。但值得注意的是，即使在60,005的双倍的词汇量下，TibLex-Yinyi分词器仍能生成显著短于子词基线的分词序列。由此可得结论：在BERT、RoBERTa等典型词汇量不超过60,000的实际应用场景中，TibLex分词器的效率优势仍具有普适性。

	TNCC（Title TC）	TNCC（LTC）
词表规模= 30005		
Subword(Tibert)	324.6	17.8
TibLex-Yinyi-NoIndex	196.6	13.2
词表规模= 45005		
Subword(Tibert)	257.3	13.3
TibLex-Yinyi-NoIndex	190.2	12.4
词表规模= 60005		
Subword(Tibert)	243.4	12.6
TibLex-Yinyi-NoIndex	187.2	12.1

Table 7: 不同词表规模下的分词器平均分词序列比较

5 总结与展望

本研究提出TibLex分词方法，并通过系统实验验证其相对于现有分词技术的优势。相较于直接应用子词分词的藏文处理方案，TibLex分词器不仅在下游自然语言理解任务中展现出竞争力，更重要的是具备显著效率优势与鲁棒性。通过系列消融实验，我们深入解析了TibLex分词高效性的成因。鉴于TibLex分词的显著优势，我们认为其可成为现有藏文分词方案的更优替代，这种效率优化显著降低模型训练与推理的算力需求，有助于减少使用这些大型语言模型的环境成本，进一步迈向绿色人工智能，尤其在效率与鲁棒性要求苛刻的应用场景中。本方法可能拓展至其他形态学特征稀疏的语言，并有望衍生出更复杂的分词优化方法，以及在处理具有真实世界噪声的输入的实际问题，相关探索将留待未来研究。

致谢

本论文得到了国家社科基金(22&ZD035)和中国工程院科技战略咨询项目(2025-XZ-16-06)，以及国家自然科学基金(61972436)和中央民族大学项目(2025XYCM39)的资助。

参考文献

Emily Bender. 2019. The# benderrule: On naming the languages we study and why it matters. *The Gradient*, 14:34.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2023. Spanish pre-trained bert model and evaluation data. *arXiv preprint arXiv:2308.02976*.

Shaosheng Cao, Wei Lu, Jun Zhou, and Xiaolong Li. 2018. cw2vec: Learning chinese word embeddings with stroke n-gram information. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

David Chandler, David Chapman, Robert Chilton, Tony Duff, Chris Fynn, Nathaniel Grove, David Germano, Steve Hartwell, Peter Hauer, and Andrew West. 2004. Thl extended wylie transliteration scheme. Working Draft. A collaborative project of the Tibetan and Himalayan Library (THL).

Aditi Chaudhary, Chunting Zhou, Lori Levin, Graham Neubig, David R Mortensen, and Jaime G Carbonell. 2018. Adapting word embeddings to new languages with morphological and phonological subword representations. *arXiv preprint arXiv:1808.09500*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

David Germano and Nicolas Tournadre. 2010. Thl simplified phonetic transcription of standard tibetan. Working Draft. Copyright © 2003 by David Germano, Nicolas Tournadre, and THL.

Li Huang, Junjie Li, Weiwei Jiang, Zhiyu Zhang, Minchuan Chen, Shaojun Wang, and Jing Xiao. 2021. Phmospell: Phonological and morphological knowledge guided chinese spelling check. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5958–5967.

Mario Michael Krell, Matej Kosec, Sergio P. Perez, and Andrew Fitzgibbon. 2021. Efficient sequence packing without cross-contamination: Accelerating large language models without impacting performance. *arXiv preprint arXiv:2107.02027*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2019. Flaubert: Unsupervised language model pre-training for french. *arXiv preprint arXiv:1912.05372*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sisi Liu, Junjie Deng, Yuan Sun, and Xiaobing Zhao. 2022. Tibert: Tibetan pre-trained language model. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2956–2961. IEEE.

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. Charbert: Character-aware pre-trained language model. *arXiv preprint arXiv:2011.01513*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Nuo Qun, Xing Li, Xipeng Qiu, and Xuanjing Huang. 2017. End-to-end neural text classification for tibetan. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data: 16th China National Conference, CCL 2017, and 5th International Symposium, NLP-NABD 2017, Nanjing, China, October 13-15, 2017, Proceedings 5*, pages 472–480. Springer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Chenglei Si, Zhengyan Zhang, Yingfa Chen, Fanchao Qi, Xiaozhi Wang, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2023. Sub-character tokenization for chinese pretrained language models. *Transactions of the Association for Computational Linguistics*, 11:469–487.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Ziqing Yang, Zihang Xu, Yiming Cui, Baoxin Wang, Min Lin, Dayong Wu, and Zhigang Chen. 2022. Cino: A chinese minority pre-trained language model. *arXiv preprint arXiv:2202.13558*.

Yun Zhang, Yongguo Liu, Jiajing Zhu, Ziqiang Zheng, Xiaofeng Liu, Weiguang Wang, Zijie Chen, and Shuangqing Zhai. 2019. Learning chinese word embeddings from stroke, structure and pinyin of characters. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1011–1020.

吉加本. 2013. 藏文文法《三十颂》研究. Ph.D. thesis, 青海师范大学.

孙媛, 刘思思, 陈超凡, 旦正错, and 赵小兵. 2024. 面向机器阅读理解的高质量藏语数据集构建. 中文信息学报, 38(3):56–64.

瞿霭堂and 劲松. 2011. 《音势论》和藏文创制的原理. 民族语文, (5):11.

附录. TibLex与SentencePiece分词方法对比示例

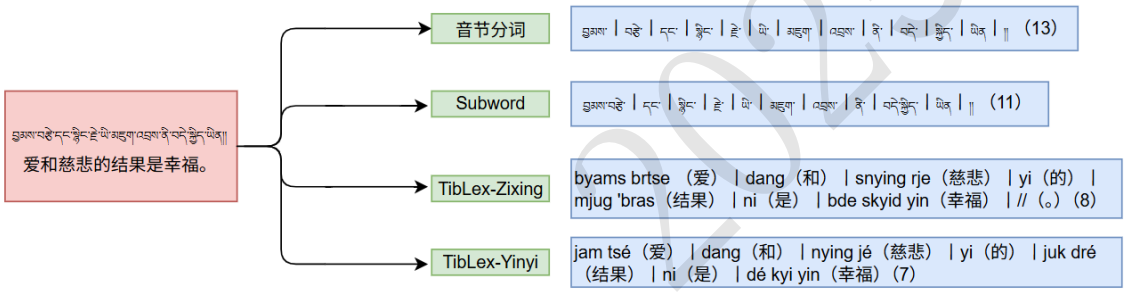

Figure 7: 不同分词方法对比：TibLex 与SentencePiece

图7展示了输入序列经过不同分词方法处理后的结果对比。从图中可以看出，TibLex 能有效减少输入序列的token 数量（TibLex-Yinyi:由11 个减少至7 个），显著提升了序列压缩效率。这种压缩优势有助于降低模型的计算成本，并在预训练和微调过程中提升训练效率。