Ko-LLaMA:基于LLaMA的朝鲜语大语言模型

庞杰[1,2,3]　闫晓东[1,2,3,*]　赵小兵[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民族语言中心
[3]民族语言智能与安全治理教育部重点实验室
*通讯作者：闫晓东
{22302041,yanxiaodong}@muc.edu.cn,nmzxb_cn@163.com

摘要

大语言模型在这两年受到了非常广泛的关注，像ChatGPT和GPT-4这样的大型语言模型（LLMs）极大地改变了自然语言处理研究，并在通向人工通用智能（AGI）的道路上迈出了令人兴奋的步伐。尽管已经开源了LLaMA等几个大型语言模型，但这些模型主要关注英文和中文语料库，对其他语言的适用性有限。而对于少数民族语言如朝鲜语来说，大语言模型的适用性更加有限。在本文中，我们通过扩展LLaMA现有的词表，增加了额外的20,000个朝鲜语Token，从而提高了其对朝鲜语的编码和语义理解的能力；并且进一步使用朝鲜语数据进行继续预训练，使用朝鲜语指令微调数据集对模型进行SFT（Supervised Fine-Tuning），并分析了不同数据量对指令精调效果的影响，经过继续预训练和指令微调后的模型显著提高了理解和遵循朝鲜语指令的能力。通过上述训练，极大增强了LLaMA的理解和生成朝鲜语文本的能力，并增强了其遵循指令的能力。实验结果表明，新提出的模型Ko-LLaMA显著提高了原版LLaMA在理解和生成朝鲜语内容方面的能力。此外，在鲜语文本分类数据集YNAT上对Ko-LLaMA与擅长少数民族语言的CINO模型及CINO的多种模型组合以及原版LLaMA和GPT-3.5进行了效果对比。结果表明，Ko-LLaMA的朝鲜语文本分类能力远超CINO和CINO的组合模型以及LLaMA和GPT-3.5等未经过朝鲜语语料进行词表扩充和继续预训练的大语言模型。

关键词： 朝鲜语；大语言模型；词表扩充；继续预训练；指令微调

Ko-LLaMA: A Korean Large Language Model Based on LLaMA

Jie Pang[1,2,3]　Xiaodong Yan[1,2,3,*]　Xiaobing Zhao[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]Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance of MOE
Corresponding author: Xiaodong Yan
{22302041, yanxiaodong}@muc.edu.cn, nmzxb_cn@163.com

Abstract

Large language models have gained immense popularity in the last couple of years, with models like ChatGPT and GPT-4 revolutionizing natural language processing research and taking exciting steps towards artificial general intelligence (AGI). Despite several large language models being open-sourced, such as LLaMA, these models primarily focus on English and Chinese corpora, with limited applicability to other languages. For minority languages such as Korean, the applicability of large language models is even more limited.In this paper, we enhance the applicability of LLaMA to the Korean language by extending its existing vocabulary with an additional 20,000 Korean tokens, improving its ability to encode and semantically understand Korean. We further

continue pre-training the model with Korean data, fine-tune the model with a Korean instruction dataset (SFT: Supervised Fine-Tuning), and analyze the impact of varying amounts of data on the fine-tuning effect. The model after continued pre-training and instruction fine-tuning significantly improves the model's ability to understand and execute Korean instructions. With the proposed approach, the capability of LLaMA to understand and generate Korean text is greatly enhanced, and its ability to follow instructions is strengthened. Experimental results show that the newly proposed model, Ko-LLaMA, significantly outperforms the original LLaMA in terms of understanding and generating Korean content. Furthermore, in the comparison of effectiveness on the YNAT dataset for fresh language text classification, Ko-LLaMA was compared against the CINO model, which excels in minority languages, along with various combinations of CINO models, original LLaMA, and GPT-3.5. The results indicate that Ko-LLaMA's ability in classifying Korean text far surpasses that of CINO and its combinations, as well as LLaMA and GPT-3.5, which have not undergone vocabulary expansion and continued pre-training on Korean language corpora.

Keywords: Korean , Large language model , Vocabulary extension , Continued pretraining , Command fine-tuning

1 引言

随着大型语言模型（LLMs）的出现，自然语言处理领域经历了实质性的范式转变。这些模型以其庞大的规模和全面的训练数据而受到关注，它们在理解和生成类似人类的文本方面表现出了非凡的能力。与专注于文本理解的预训练语言模型（如BERT）不同，GPT系列(Brown et al., 2020)强调了文本生成，使它们相比其他模型更适合发挥创造性。值得注意的是，GPT家族的最新成员，即ChatGPT和GPT-4，受到了极大的关注，他们在这个迅速发展的领域中确立了自己的领先地位。

然而，尽管LLMs具有很大影响力，但LLMs的实施具有很大的限制，这些限制阻碍了透明和开放的研究。最主要的问题是他们的专有性质，限制了对模型的访问，从而阻碍了更广泛的研究社区基于他们的成功进行建设。此外，训练和部署这些模型所需大量计算资源，对资源有限的大多数研究者来说是个挑战，进一步加剧了可研究性的问题。为了应对这些限制，NLP研究倾向于使用开源替代品，以增加更大的透明度和协作。LLaMA(Touvron et al., 2023a)、LLaMA-2(Touvron et al., 2023b)和Alpaca(San-Martin et al., 1968)就是这些倡议的显著例子。这些开源的LLMs旨在促进学术研究，加快NLP领域的进步。开源这些模型的目的是创造一个有利于模型开发、微调和评估的环境，最终创建适用于各种用途强大、有能力的LLMs，所以为了弥补少数民族语言朝鲜语在大语言模型上的空缺，我们研究一款可以用于少数民族朝鲜语的的大语言模型，便于以后朝鲜语大语言模型的研究与发展。

目前,中英文领域的大语言模型研究已经取得了很好的发展,而对于少数民族语言如朝鲜语来说,大语言模型仍处于萌芽阶段，没有发挥出应有的效果。中英文领域的大语言模型的词汇表几乎没有朝鲜语Token，无法编码和解码朝鲜语文本。通过对原版LLaMA模型进行词表扩充、继续预训练、指令微调等工作，极大的提高了LLaMA对朝鲜语的理解和生成的能力，充分挖掘了大语言模型在朝鲜语的能力。

本文的主要贡献如下：

1) 通过收集大量的朝鲜语训练语料，使用SentencePiece工具采用BPE的分词方式对语料进行分词，并与原版LLaMA词表进行合并，扩充了20,000个朝鲜语Token。并且采用低秩适配(LoRA)方法，对进行词表扩充后的模型进行了继续预训练，在使用较少计算资源的情况下显著提高了模型在朝鲜语的理解和生成能力。

2) 通过公开的YNAT朝鲜语文本分类数据集构造了包括45678条训练集、9104条验证集，9104条测试集的朝鲜语文本分类的指令微调数据集，并对模型进行指令微调，

分析了不同数据量对模型指令微调效果的影响，显著提高了模型朝鲜语文本分类的能力。通过在YNAT朝鲜语文本分类数据集上的对比实验，表明Ko-LLaMA的效果超过CINO及CINO相关组合模型。

2 相关工作

大语言模型在这两年取得了巨大的进步，接下来将现梳理大语言模型近几年的研究进展，再介绍词表扩充、继续预训练、指令微调一系列训练范式的发展状况。

OpenAI在2018年首次提出了GPT(Generative Pretrained Transformer)(Radford et al., 2018)。这种模型使用了Transformer的解码器架构，并使用了一个单向的语言模型目标进行预训练。在预训练后，GPT可以通过在特定任务的数据上进行微调来适应各种NLP任务，其工作原理与Bert类似，当作预训练模型使用。在2019年，OpenAI发布了GPT的第二版，即GPT-2(Radford et al., 2019)。与GPT相比，GPT-2有更多的参数（从1.1亿增加到3.4亿）。GPT-2在许多NLP基准测试中取得了领先的表现。2020年，OpenAI发布了GPT-3(Mann et al., 2020)，是当时最大的语言模型，拥有1750亿个参数。GPT-3在文本生成任务上的表现超越了许多先前的模型。在GPT-3的基础上，OpenAI进一步进行了微调，于2021年发布了InstructGPT(Ouyang et al., 2022)。这个模型经过了大量的模型训练和数据清洗，旨在理解和执行用户的指令。InstructGPT的训练过程包括两个阶段。第一阶段是预训练，这是在大量的互联网文本上进行的。而第二阶段是微调，这是在一个更小，特定的，由人类审核员生成的数据集上进行的。InstructGPT在许多实际应用中都表现出色，如进行技术支持、提供教育资源、帮助用户学习新技能等。由于InstructGPT的成功，OpenAI受到启发并于2022年11月底发布了ChatGPT，ChatGPT一经发布便受到了广泛的关注，迅速火遍全网。InstructGPT和ChatGPT都是从GPT-3微调而来，但他们的区别是，InstructGPT是为了理解和执行用户的指令，而ChatGPT则是为了进行自由形式的对话。基于上述区别，确定了朝鲜语大语言模型要先经过InstructGPT的训练阶段，但是由于GPT没有开源，但是上述工作要在开源的模型上展开，所以LLaMA(Touvron et al., 2023a)作为Meta开源的模型则是一个比较好的选择，而且LLaMA在各项NLP任务上取得了显著的成果。

LLaMA作为一款开源的大语言模型在大语言模型研究上做出了突出的贡献，由于LLaMA只在英文的效果上表现卓越，在其他语种上展现出来的能力并没有很突出，所以LLaMA在其他语言上的加固和拓展也基本采用了词表扩充+继续预训练+指令微调的训练范式。Chinese-LLaMA(Cui et al., 2024)是在LLaMA的基础上增强了中文能力，这也提供了一种思路。虽然加固其中文能力比开发出LLaMA的朝鲜语理解和生成的能力要简单一些，但是验证了这一思路的可行性。

3 Ko-LLaMA训练流程

3.1 原版LLaMA-7B词表扩充

LLaMA-7B的训练集大约包含1.4T的Token，其中大部分是英文，一小部分是拉丁语和其他欧洲语言(Xia et al., 2023)。因此，LLaMA几种语言上的能力表明了它具有多语种和跨语种理解能力。初步研究表明，LLaMA几乎没有朝鲜语理解和生成的能力，在朝鲜语的各类NLP任务上表现都有待提高。为了赋予LLaMA更强的朝鲜语理解和生成能力，我们使用朝鲜语语料库对LLaMA模型进行继续预训练。然而，直接使用朝鲜语语料库进行继续预训练面临一些挑战。首先，原始的LLaMA词汇表包含的朝鲜语Token非常少，这不足以编码一般的朝鲜语文本。虽然LLaMA Tokenizer可以通过回退到字节码来支持所有的朝鲜语字符，但这种回退策略会显著增加序列长度，因为每个朝鲜语字符都被拆分为3-4个字节Token，降低了朝鲜语文本的编码和解码效率。其次，字节Tokens不仅用于表示朝鲜语字符，还用于表示其它UTF-8 Tokens，所以使用字节Tokens的方法很难让LLaMA有效地学习捕获朝鲜语Token语义的表示。

为了解决这些问题并提高编码效率，我们使用额外的朝鲜语Token扩展LLaMA词表，并调整模型以适应扩展后的词表(Gao et al., 2023)。扩展过程如下：

1) 为了增强分词器对朝鲜语文本的支持，首先使用SentencePiece(Kudo and Richardson, 2018)采用BPE的方式在朝鲜语语料库上训练一个朝鲜语分词器，词汇量为20,000。

2) 随后，通过合并他们的词汇表，将朝鲜语分词器合并到原版LLaMA分词器中。这样就可以得到了一个合并过的分词器，就是包含了朝鲜语的LLaMA分词器，它的词汇量为49924。

3) 为了使LLaMA模型适应朝鲜语LLaMA分词器，我们将词嵌入层和语言模型头从V×H形状调整为V'×H，其中V=32000表示原始词汇表的大小，H表示词嵌入的维度，V' = 49924是朝鲜语LLaMA分词器的新词汇表大小。新添加的行被添加到原始嵌入矩阵的末尾，确保原始词汇表中的标记的嵌入不受影响。

	Length	Content
原始朝鲜语句子	80	최근 길림 연변변경관리지대 경신변경파출소 경찰은 영상순찰을 하던 중 야생동북범 한마리가 대도천촌 부근의 숲속에서 먹이를 찾는 모습을 발견했다.
原版LLaMA Tokenizer	137	['▁', '<0xEC>', '<0xB5>', '<0x9C>', '<0xEA>', '<0xB7>', '<0xBC>', '▁', '<0xEA>', '<0xB8>', '<0xB8>', '<0xEB>', '<0xA6>', '<0xBC>', '▁', '연', '<0xEB>', '<0xB3>', '<0x80>', '<0xEB>', '<0xB3>', '<0x80>', '경', '<0xEA>', '<0xB4>', '<0x80>', '리', '지', '대', '▁', '경', '신', '<0xEB>', '<0xB3>', '<0x80>', '경', '<0xED>', '<0x8C>', '<0x8C>', '<0xEC>', '<0xB6>', '<0x9C>', '소', '▁', '경', '<0xEC>', '<0xB0>', '<0xB0>', '은', '▁', '영', '상', '<0xEC>', '<0x88>', '<0x9C>', '<0xEC>', '<0xB0>', '<0xB0>', '을', '▁', '하', '<0xEB>', '<0x8D>', '<0x98>', '▁', '중', '▁', '<0xEC>', '<0x95>', '<0xBC>', '<0xEC>', '<0x83>', '<0x9D>', '동', '<0xEB>', '<0xB6>', '<0x81>', '<0xEB>', '<0xB2>', '<0x94>', '▁', '한', '마', '리', '가', '▁', '대', '도', '천', '<0xEC>', '<0xB4>', '<0x8C>', '▁', '부', '<0xEA>', '<0xB7>', '<0xBC>', '의', '▁', '<0xEC>', '<0x88>', '<0xB2>', '<0xEC>', '<0x86>', '<0x8D>', '에', '서', '▁', '<0xEB>', '<0xA8>', '<0xB9>', '이', '를', '▁', '<0xEC>', '<0xB0>', '<0xBE>', '는', '▁', '모', '<0xEC>', '<0x8A>', '<0xB5>', '을', '▁', '<0xEB>', '<0xB0>', '<0x9C>', '<0xEA>', '<0xB2>', '<0xAC>', '<0xED>', '<0x96>', '<0x88>', '다', '.', '▁']
Ko-LLaMA Tokenizer	35	['▁최근', '▁길림', '▁연변', '변경', '관리', '지대', '▁경신', '변경', '파출소', '▁경찰', '은', '▁영상', '순찰', '을', '▁하', '던', '▁중', '▁야생동', '북범', '▁한마', '리가', '▁대', '도천촌', '▁부근', '의', '▁', '숲', '속에', '서', '▁먹', '이를', '▁찾는', '▁모습을', '▁발견했다', '.']

图 1 原始LLaMA和Ko-LLaMA的tokenizer的效果比较

3.2 使用LoRA对LLaMA进行继续预训练

常规的大语言模型训练范式即更新LLMs的全量参数是非常昂贵的，并且在时间和成本上对大多数实验室或公司来说是不可行的。低秩适配（LoRA）(Hu et al., 2021)是一种参数高效的训练方法，它保持预训练模型的权重不变，同时引入可训练的秩分解矩阵。LoRA冻结预训练模型的权重，并在每一层注入可训练的低秩矩阵。这种方法显著减少了总的可训练参数，使得用更少的计算资源训练LLMs成为可能。具体来说，对于一个线性层，其权重矩阵$W_0 \in R^{d*k}$，其中k是输入维度，d是输出维度，LoRA添加了两个低秩分解的可训练矩阵$B \in R^{d*r}$和$A \in R^{r*K}$，其中r是预先设定的秩。带有输入x的前向传递公式如下:

$$h = W_0x + \triangle Wx = W_0x + BAx, B \in R^{d*r}, A \in R^{r*k} \tag{1}$$

在训练过程中，W_0是冻结的，不接收梯度更新，而B和A是可以更新的。通过选择秩rmin(d, k)，我们减少了内存消耗，因为我们不需要为大的冻结矩阵存储优化器状态。我们主要将LoRA适配器集成到注意力模块和MLP层的权重中，因为在QLoRA(Dettmers et al., 2023)中验证了将LoRA应用到所有线性Transformer块的有效性，表明这种选择是合理的。

我们使用标准的因果语言模型（CLM）任务对LLaMA进行朝鲜语上的继续预训练，给定输入序列$x = (x_0, x_1, x_2, ...)$，模型采用自回归的方式预测下一个Token，目标就是最小化负对数似然:

$$L_{CLM}(\Theta) = -\sum_i logp(x_i|x_0, x_1, , , , , x_{i-1}; \Theta) \tag{2}$$

经过继续预训练以后模型能够完成对已输入文本进行续写，即提高了模型对朝鲜语文本的理解和生成能力，接下来只需要对继续预训练以后的模型做任务上的指令微调就能让模型完成具体的下游任务。

3.3 朝鲜语指令微调

由于模型学习的知识是非常广泛的，所以我们对模型的输入输出难以控制，并且经常会生成一些无关的内容，这是因为公式(2)中的语言建模目标是预测下一个词而不是'follow instructions with human feedback.'(Ouyang et al., 2022)。为了使语言模型的行为与用户的意图相符，可以对模型进行微调，明确地训练它遵循指示。斯坦福Alpaca(Taori et al., 2023)是一个基于LLaMA的指令模型，它是在由Self-Instruct(Wang et al., 2023)中的技术生成的52K条指令数据上进行训练的。我们遵循斯坦福Alpaca的方法，对朝鲜语LLaMA进行有监督的微调训练，以训练出一个指令遵循模型。

由于数据有限，没有对继续预训练以后的LLaMA做大规模的指令微调，所以只挑选了一个NLP常见的任务：文本分类，挑选数据进行指令微调，并且与CINO(Yang et al., 2022)和CINO的一些组合模型以及原版LLamA和GPT-3.5等大语言模型做效果比对。

在进行指令微调数据之前，需要明确两个事情，一个是Prompt模版如何确定，另外一个是指令微调的数据从何而来。由于Chinese-LLaMA-Alpaca(Cui et al., 2023)已经成功进行了指令微调，所以这里的Prompt模版还是采用斯坦福Alpaca模版，不过因为我们是要做朝鲜语的指令微调，所以要将其格式转换成朝鲜语版本。具体Prompt形式如下图2所示：

아래는 작업을 설명하는 명령입니다.요청을 적절하게 완료하는 응답을 작성합니다. ###지시: {instruction} ###반응:{output}	下面是描述任务的指令。编写适当完成指令的回复。 ###指令: {instruction} ###回复:{output}

图 2 指令微调的Prompt格式

通过使用上述Prompt模版进行构造数据，并且使用公开数据集YNAT经过数据扩充来构造指令微调的数据，YNAT公开的朝鲜语文本分类数据集，其中训练集45678条，验证集9104条，测试集9104条，并且将最终结果与CINO模型及CINO相关组合模型进行结果比较；同时为了验证与相关大模型在朝鲜语上理解能力的差距，设置了Ko-LLaMA与原版LLaMA和GPT-3.5在朝鲜语文本分类数据集上的对比实验。具体结果分析放在实验与分析环节。

4 实验与分析

4.1 继续预训练实验

4.1.1 实验数据集

继续预训练的数据集主要是从新华网、人民网朝鲜语版等多家新闻媒体网站上爬取58642篇新闻文本作为训练集原始语料以及230篇新闻文本作为测试集原始语料。将部分语料爬取以后经过去除乱码、特殊符号、图片等信息过滤得到了比较纯净的新闻文本，共计680MB。

4.1.2 模型评估标准

PPL（困惑度）是一种衡量语言模型质量的评价指标。它衡量模型预测下一个单词的难度。PPL 越低，模型越好。具体来说，PPL 是给定数据集上模型预测的单词序列的对数似然函数的负值。对于一个包含N 个单词的数据集，PPL 计算如下：

$$PPL = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i|w_1,...,w_{i-1})\right) \tag{3}$$

因为LLaMA在经过词表扩充和继续预训练以后，已经可以完成基本的文本续写，并且其训练预料大多是新闻文本，在朝鲜语新闻文本续写效果最为明显，所以对继续预训练以后的模型在230篇新闻文本上采用困惑度作为评估指标进行评估，这里需要注意一点，针对同一个测试集，采用不同tokenizer的模型会导致公式（3）中N的大小不同，即序列长度不同，所以tokenizer不同的模型进行PPL值的对比是没有意义的，因此本文采用词表扩充未经过继续预训练的模型和词表扩充并经过继续预训练的模型在230篇新闻文本续写任务的PPL值进行对比，这样保证tokenizer的一致性，也就是N相同，进而可以描述出模型经过继续预训练以后对朝鲜语的理解和生成能力的提升。

4.1.3 实验设置

在继续预训练部分，由于数据集中没有其他大模型训练时用的那么多语料，并且原版LLaMA模型基本没有朝鲜语的编码和解码能力，所以我们设置的学习率等参数相对大一些便于模型快速学习朝鲜语语料的知识，下面表1为继续预训练的详细参数设置。

模型参数	参数值
Peak learning rate	$2e-4$
LoRA dropout	0.05
Batch size	1024
Max sequence len	512
LoRA rank	8
Torch dtype	Float 16
LoRA alpha	32
Weight decay	0.01

表 1 继续预训练的参数设置

在继续预训练阶段我们对模型进行了全面的预训练，包括Embedding层，这样不仅可以使模型适应新添加的朝鲜语Token，同时可以最小化对原始模型的干扰，并且我们向模型添加了LoRA权重（adapter），使用PEFT库去做带有LoRA的参数高效训练，同时使用DeepSpeed(Aminabadi et al., 2022)来优化训练过程中的内存效率，采用AdamW优化器(Loshchilov and Hutter, 2019)，峰值学习率为2e-4，warm-up cosine scheduler为5%。此外，我们还使用值为1.0的梯度裁剪，以缓解潜在的梯度爆炸。最终在两张A100 GPU（80GB VRAM）进行了一次5小时的迭代训练。

4.1.4 实验结果分析

本节使用LLaMA-7B模型经过朝鲜语词表扩充以后做继续预训练，分别在爬取的训练数据上进行预训练，其中58642篇新闻文本按照9:1的比例划分训练集和验证集，测试集230篇新闻文本，验证集Loss下降图如图3所示。继续预训练阶段的Loss是前期急剧下降，然后变得平缓并趋于稳定，对此做出的分析是，在预训练阶段模型并不具备朝鲜语的编解码能力，模型参数还没有得到很好的调整，误差相对较大，因此每次参数更新都在较大程度上减小误差，所以损失下降得快；随着训练的进行，模型参数逐渐接近最优解，每次参数更新带来的误差减小，因此Loss下降速度变慢。

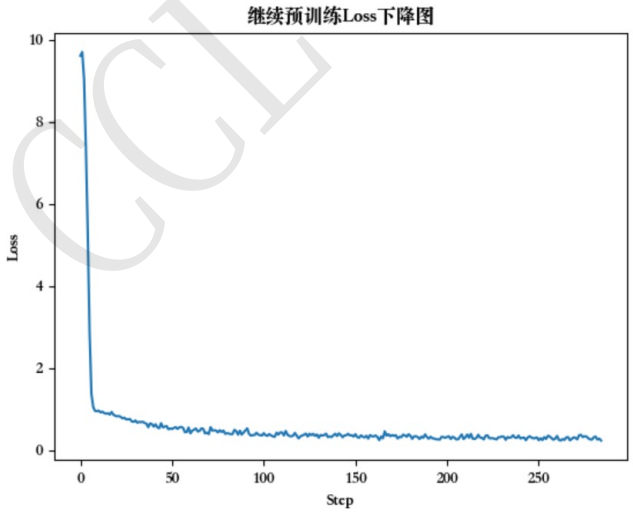

图 3 继续预训练Loss下降图

由于需要考虑PPL可比性的问题，所以在PPL对比实验中需要保证对比模型的词表一致，为了说明继续预训练对Ko-LLaMA的朝鲜语理解和生成能力的提升，我们将扩充词表但是没有经过继续预训练的模型和扩充词表并经过继续预训练的模型在230篇新闻文本上的续写任务计算得到的PPL值进行对比。在文本续写中，扩充词表但是没有经过继续预训练的模型的PPL值为41.287，扩充词表并经过训练以后的模型的PPL值为11.683，具体结果如下表2所示。这表明

继续预训练以后的模型能够有效地学习新闻文本中的语言模式并预测单词序列。下图4为扩充词表并经过继续预训练模型的续写效果展示。

	扩充词表未进行继续预训练模型	扩充词表经过继续预训练模型
PPL值	41.287	11.683

表 2: 继续预训练前后模型的PPL值对比

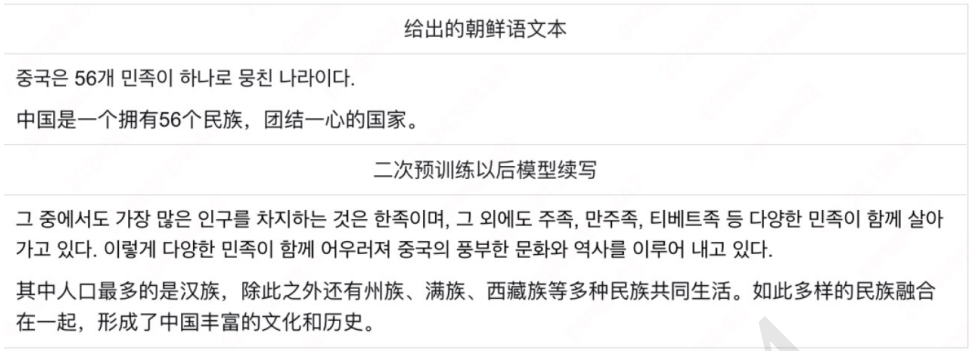

图 4 经过继续预训练后模型的续写效果

我们最终得出结论：经过预训练后的模型在文本建模任务上的良好性能表明，它可以用于各种自然语言处理应用，例如文本摘要、机器翻译和问答以及分类。进一步的研究可以探索该模型在各种数据集和任务上的性能，并对其进行微调以提高特定任务的性能。

4.2 指令微调实验

4.2.1 实验数据集

指令微调的训练数据是直接通过公开的YNAT数据按照斯坦福的Alpaca格式的Prompt模版进行拼接得到，其中训练集45678条，验证集9104条，测试集9104条，包含朝鲜语文本及其所处的类别共七个主题科技、经济、文化、美容/健康、社会、生活、世界。数据样例见图5。

Prompt	类别
아래는 작업을 설명하는 명령입니다.요청을 적절하게 완료하는 응답을 작성합니다. ###지시: 다음 텍스트에 대한 분류 별로 각각은 ["정치", "경제", "사회","문화","세계","IT\스포츠 과학적",""]. 실업급여 신청은 느는데 고용은 끊겨 ...현실적인 고용 위기다 ###반응:사회	사회
译文	
下面是描述任务的指令。编写适当完成指令的回复。 ###指令: 对下列文本进行分类，类别分别是["政治"、"经济"、"社会''、"文化"、"世界"、"IT\科学"、"体育"]。 失业补贴的申请在增加，而招聘却中断....这是一场现实中的雇佣危机 ###回复:社会	社会

图 5 朝鲜语文本分类Prompt样例

4.2.2 模型评估标准

本节在Ko-LLaMA与CINO及相关模型进行分类效果评定的时候标准采用准确率(A)、精确率(P)、召回率(R)以及F1值作为实验的评价标准.计算公式为:

$$A = \frac{T_{TP} + T_{TN}}{T_{TP} + T_{TN} + T_{FP} + T_{FN}} \tag{4}$$

$$P = \frac{T_{TP}}{T_{TP} + T_{FP}} \tag{5}$$

$$R = \frac{T_{TP}}{T_{TP} + T_{FN}} \tag{6}$$

$$F1 = \frac{2 * P * R}{P + R} \tag{7}$$

式中：T_{TP}为样本属于类C_i，并被分类器正确分类到类C_i的样本数；T_{FP}为样本不属于类C_i，但被分类器分到类C_i的样本数；T_{FN}为样本属于类C_i，但被分类器分到其他类的样本数。本文所研究的朝鲜语文本分类属于多分类任务，因此采用宏平均来衡量整体的分类效果,即单独计算每个类别的P和R，再进行算术平均得到测试集总体的P和R,最后通过(7)式得到F1。

4.2.3 实验设置

指令微调部分参数设置与继续预训练部分参数设置是不同的，因为在指令微调的时候，模型已经学习过朝鲜语知识，并且我们主要的任务是对齐问答格式，所以我们可以设置的相对小一些。下面的表2是指令微调时设置的参数详情。

模型参数	参数值
Peak learning rate	$1e-4$
LoRA dropout	0.05
Batch size	1024
Max sequence len	512
LoRA rank	8
Torch dtype	Float 16
LoRA alpha	32
Weight decay	0

表 3 指令微调训练的参数设置

本节实验分为两个阶段：

1) 验证不同数据量对经过预训练以后模型能力的影响，分别随机抽取训练集数据的1%、5%、20%、50%、80%、100%的数据对继续预训练以后的模型进行指令微调并验证效果。

2) 分别采用CINO及CINO+TextCNN、CINO+BiLSTM、CINO+TextCNN+BiLSTM（模型串联，下面称为Model series）、CINO+TextCNN/BiLSTM（TextCNN和BiLSTM并联在CINO后面，下面称为Model parallel）以及Ko-LLaMA在YNAT数据集上进行评测实验。

4.2.4 实验结果分析

指令微调阶段的验证集Loss如下图6所示。和预训练阶段Loss下降非常相似，都是前期急剧下降，然后变得平缓并趋于稳定。对此做出的分析：

虽然模型已经在大量数据上预训练过，但是当模型开始在特定任务的数据上进行微调时，它仍然需要进行一些参数的调整以适应这个新任务。在微调的初始阶段，模型由于对新任务的数据分布不熟悉，因此可能会有较大的误差，这就导致了在微调初期下降得较快。而随着微调

的进行，模型对新任务的数据分布逐渐熟悉，每次更新带来的误差减小，因此Loss下降的速度会变慢。

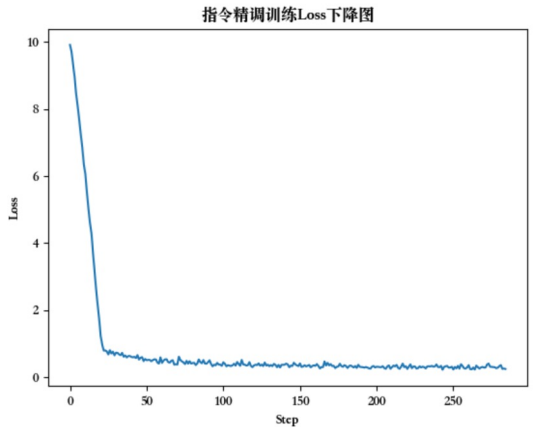

图 6 指令微调Loss损失图

第一阶段的实验：验证不同数据量对预训练以后模型能力的影响，所以我们分别随机抽取了训练数据的1%、5%、20%、50%、80%、100%进行微调，其F1-score的变化如下图7所示。

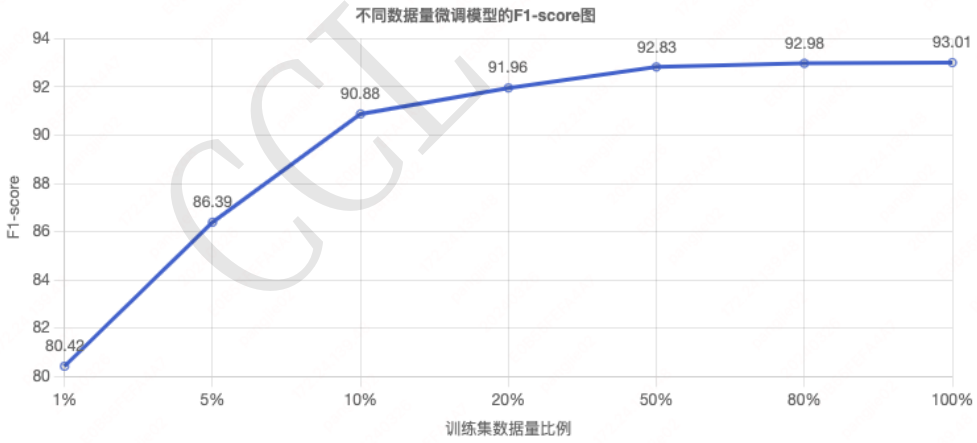

图 7 不同数据量指令微调模型的F1-score图

从图8中可以看出不同数据量对应的F1-score值的变化情况符合Loss下降的规律，当仅用1%的数据时，模型并没有很好的被充分训练，所以F1-score低，Loss高；当继续增加训练数据时，Loss急剧下降，F1-score迅速升高；再继续增加数据至50%时，F1-score就已经趋于平稳，再继续增加数据的收益将会很低。

表 4 不同模型分类效果的对比实验数据(单位:%)

模型	准确率(A)	精确率(P)	召回率(R)	F1-score
CINO	87.46	86.57	88.07	87.29
CINO+TextCNN	86.34	86.42	86.2698	86.28
CINO+BiLSTM	86.62	86.89	86.77	86.78
Model series	87.58	86.15	88.75	87.43
Model parallel	87.99	87.58	89.31	88.44
原版LLaMA	52.41	53.01	49.68	51.36
GPT-3.5	71.33	72.68	73.59	73.13
Ko-LLaMA	**92.56**	**92.83**	**93.18**	**93.01**

第二阶段实验：使用全部指令微调数据在经过在继续预训练以后的模型上做文本分类任务的指令微微调，Ko-LLaMA在朝鲜语文本分类上的能力得到了巨大的提升。具体实验结果由表3给出，根据实验结果得出以下结论:

经过继续预训练和指令微调的Ko-LLaMA的效果要显著优于CINO的多种组合模型的效果，因为Ko-LLaMA模型参数更多并且已经在大规模的语料库上进行了预训练，当其经过词表扩充以及继续预训练，模型可以很快的学习到朝鲜语的相关知识并进行理解；同时Ko-LLaMA的效果也优于未经过词表扩充和继续预训练的大语言模型（原版LLaMA和GPT-3.5）的效果，对此的分析为原版LLaMA和GPT-3.5词表中缺少朝鲜语Token导致对朝鲜语的编码能力很弱，并且未经过朝鲜语语料的预训练更将导致对朝鲜语的理解能力不理想。最后在指令微调的数据集上，通过特定的格式的Prompt引导模型如何完成一个分类任务，这样模型可以更好的学习到如何去完成一个分类任务。最终Ko-LLaMA在YNAT朝鲜语文本分类数据集上的效果为：准确率（A）92.57%、精确率（P）92.83%、召回率（R）93.19%、F1-score 93.01%。

5 总结

本文提出了一种基于LLaMA并且能够理解少数民族语言-朝鲜语的大语言模型Ko-LLaMA。因为原版LLaMA无法理解和生成朝鲜语，所以我们增加了额外的2万个朝鲜语Token，并且与原版LLaMA进行了词表合并，最终词表大小为49924；然后通过爬取的58642篇朝鲜语文档进行了继续预训练，在训练过程中使用PEFT库去做带有LoRA的参数高效训练，同时使用DeepSpeed来优化训练过程中的内存效率，通过继续预训练，使Ko-LLaMA能够理解和生成朝鲜语，由于数据量的限制，朝鲜语的的理解和生成能力并不是特别强，只能进行一些基本的续写；然后我们通过公开数据集YNAT，按照斯坦福Alpaca的Prompt格式进行构造数据，并且对继续预训练以后的模型进行指令微调训练，通过随机抽取不同比例的指令微调数据进行指令微调训练，分析了不同数据量对指令微调效果的影响，经过全部数据指令微调训练以后的Ko-LLaMA在朝鲜语文本分类任务上的能力已经得到了显著提高。我们通过与CINO、CINO+TextCNN、CINO+BiLSTM、CINO+TextCNN+BiLSTM、CINO+TextCNN/BiLSTM以及原版LLaMA和GPT-3.5等一系列模型进行比较，Ko-LLaMA的朝鲜语文本分类效果比CINO等一系列模型显著提高。

致谢

本论文由国家社科基金重点项目(22&ZD035)，中央民族大学项目(GRSCP202316, 2023QNYL22，2024GJYY43)资助。

参考文献

Reza Yazdani Aminabadi, Samyam Rajbhandari, Minjia Zhang, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Jeff Rasley, Shaden Smith, Olatunji Ruwase, and Yuxiong He. 2022. Deepspeed inference: Enabling efficient inference of transformer models at unprecedented scale.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Yiming Cui, Ziqing Yang, and Xin Yao. 2024. Efficient and effective text encoding for chinese llama and alpaca.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. 2023. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Ben Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

M San-Martin, M Copaira, J Zuniga, R Rodreguez, G Bustinza, and L Acosta. 1968. Aspects of reproduction in the alpaca. *Reproduction*, 16(3):395–399.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*.

Ziqing Yang, Zihang Xu, Yiming Cui, Baoxin Wang, Min Lin, Dayong Wu, and Zhigang Chen. 2022. CINO: A Chinese minority pre-trained language model. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3937–3949, Gyeongju, Republic of Korea, October. International Committee on Computational Linguistics.