



北京大学

本科生毕业论文

PolicyBERT: 基于全词掩码的中文政策文本预训练语言模型

PolicyBERT: A Chinese Policy Text Pre-trained Language Model Based on Whole Word Masking

姓名：浦皓天

学号：2100016620

院系：信息管理系

专业：大数据管理与应用

导师姓名：孟凡

二〇二五年五月

摘要

关键词：

Abstract

Keywords:

目录

1 引言 5

2 研究现状 7

3 PolicyBERT 7

3.1 研究方法 7

3.1.1 模型输入 7

3.1.2 多层融合 8

3.2 实验准备 9

3.2.1 数据集 9

3.2.2 数据预处理 10

3.2.3 微调数据集构建 11

3.2.4 骨干模型 12

3.2.5 训练配置 13

3.3 实验结果 13

3.3.1 PolicySM-1 13

3.3.2 PolicySM-2 14

4 未名问政RAG系统 17

5 结论 17

1 引言

自然语言处理（Natural Language Processing, NLP）是人工智能领域的重要分支，旨在使计算机能够理解、分析和生成自然语言。随着互联网的快速发展，海量文本数据的产生为 NLP 研究提供了丰富的资源，同时也带来了新的挑战。尤其是在中文文本处理方面，由于汉字的独特性和复杂性，传统的基于词典的方法难以满足实际需求。因此，如何有效地表示和理解中文文本成为了一个亟待解决的问题。

在自然语言处理的早期，文本表示主要依赖于基于词典的特征工程方法，如 TF-IDF 通过词频和逆文档频率量化词的重要性 [1]，主题模型（如 LDA）则挖掘文本的潜在主题分布，将文档映射到低维主题向量空间，提升了语义理解能力 [2]。随着深度学习的兴起，分布式词向量技术（如 Word2Vec [3] 和 GloVe [4]）通过大规模无标注语料训练，将词映射为低维实数向量，捕捉词间相似性，但其“静态”表示难以区分多义词，对上下文变化不敏感。为此，ELMo 引入了基于双向语言模型的上下文相关表示，使同一词在不同上下文中具有不同向量，显著提升了下游任务性能 [5]。

基于 Transformer 架构的预训练语言模型（PLM）带来了革命性变革。BERT 首创了“掩码语言模型”和“下游任务微调”范式，通过海量语料预训练并在具体任务上微调，大幅提升性能，奠定了通用 NLP 研究的基础 [6]。其后，RoBERTa 通过更大规模语料和更长训练时间刷新了多项任务表现 [7]，ERNIE 则融入知识图谱与实体信息，在预训练中引入外部知识，进一步增强了语义理解能力 [8]。

尽管上述预训练模型在英文 NLP 任务中表现优异，直接迁移到中文时却面临语言结构层面的挑战。中文语言天然缺乏空格作为词语边界标记，模型在对文本进行编码前通常需先进行分词处理。但中文的词边界存在模糊性和上下文依赖性，使得不同分词方式可能产生完全不同的语义理解。例如，“北京大学”可被切分为“北京大学”或整体作为专有名词“北京大学”，两者所表达的语义截然不同。这种分词歧义会导致 token 级别的预训练模型无法准确建模真实语义。

现有中文 BERT 模型通常采用字符级（字粒度）编码策略，即将每个汉字视为独立 token，这虽然规避了分词带来的歧义问题，但也牺牲了对词语整体语义的把握。例如，将“人工智能”拆解为“人”“工”“智”“能”，使模型需要通过更长的上下文依赖才能捕捉到其作为专有名词的统一语义。这种处理方式尤其不利于处理术语密集、长词多、句式复杂的文本类型。

在 NLP 的具体应用中，领域文本（Domain-specific Text）处理逐渐成为研究热点。尤其是法律、金融、医疗、政策等垂直领域，其文本具有术语密集、专业性强、结构复杂、长距离依赖、命名实体多、更新频繁等特点，对文本编码提出了更高要

求：模型不仅要能理解个体 token 的语义，还要具备对领域词语、命名实体的整体感知能力。这使得纯粹依赖字粒度编码的 BERT 类模型难以充分捕捉政策文本中的关键信息。若模型不能准确识别和建模“国家政策”“财政转移支付”“扶贫开发”等多字短语或专有名词，可能导致信息丢失、下游任务性能下降

ZEN (a BERT-based Chinese (Z) text encoder Enhanced by N-gram representations) 提出了引入 n-gram 表示的中文词信息 [9]。该文使用了 Shizhe Diao 等人提出的基于计算相邻字符 PMI (Pointwise Mutual Information) 的分词方法 [10]。将所有连续的 n-gram 片段与字符表示并行编码，并在模型的每一个交叉注意力组件的末尾进行相加融合，以补充多字组合信息。

本文将在 ZEN 的基础上，提出一种新的中文政策文本预训练语言模型 PolicyBERT。该模型在 ZEN 的基础上引入了 HanLP 分词、门控与注意力机制，以增强对中文政策文本的建模能力。

随着自然语言处理技术的快速发展，基于检索增强生成 (Retrieval-Augmented Generation, RAG) 的任务在政策文本处理领域展现出巨大潜力。政策文本通常具有术语密集、结构复杂、领域专有性强等特点，这对模型的理解和生成能力提出了更高要求。传统的预训练语言模型虽然在通用任务上表现优异，但在特定领域的任务中往往难以充分捕捉领域知识。因此，在子领域上微调过的预训练模型成为解决这一问题的关键，能够更好地适应特定领域的语义需求，提升 RAG 系统的性能。

Ollama 是一个高效的模型部署工具，支持本地化部署大规模语言模型，能够显著降低对外部依赖的需求。本文中，我们使用 Ollama 部署了千问 Qwen-1.5，这是一款由阿里巴巴达摩院推出的强大中文语言模型，具备卓越的生成能力和对中文语义的深度理解。此外，BGE (BAAI General Embedding) 系列模型在 RAG 任务中表现出色，尤其是 bge-base-zh-v1.5，其在中文文本嵌入生成方面具有较高的语义一致性和检索精度，为构建高效的检索模块提供了坚实基础。

在架构设计上，LongChain 是一种灵活且高效的框架，专为复杂的 RAG 系统设计，能够无缝集成检索和生成模块，支持多阶段任务处理。LongChain 的优势在于其模块化设计和对大规模模型的兼容性，使得系统能够在保持高性能的同时，具备良好的扩展性和易用性。

本文将结合 Ollama 部署的 Qwen-1.5 和微调后的 bge-base-zh-v1.5，构建一个高效的 RAG 系统。该系统将利用本地部署的 Qwen-1.5 进行文本生成，并通过使用 ZEN 架构微调后的 bge-base-zh-v1.5 实现高效的政策文本检索与生成任务。通过这

一系统，本文验证了在特定领域微调模型和优化架构设计对提升 RAG 系统性能的有效性。

本文的主要贡献包括以下几个方面：提出PolicyBert模型，在 ZEN 的基础上引入 HanLP 分词、门控与注意力机制，以增强对中文政策文本的建模能力；将 bge-base-zh-v1.5 的主体 BERT 部分参数与预训练ZEN融合，使用 ZEN 架构进行微调，以适应政策文本的特定需求；使用 Ollama 部署 Qwen-1.5，并结合 LongChain 架构构建 RAG 系统，利用本地部署的 Qwen-1.5 和微调后的 bge-base-zh-v1.5，实现高效的政策文本检索与生成任务。通过这一系统，本文验证了在特定领域微调模型和优化架构设计对提升 RAG 系统性能的有效性。本文代码开源在<https://github.com/PuHT4213/PolicyBert>上。

2 研究现状

ZEN (Pre-training Chinese Text Encoder Enhanced by N-gram Representations) 是一种专为中文设计的预训练语言模型，旨在解决中文文本处理中的分词不确定性问题 [9]。相较于将每个汉字作为独立token的传统BERT模型，ZEN引入n-gram表示，通过显式建模多粒度的词片段信息，增强了对词级语义的建模能力。在架构上，ZEN在字符级BERT的基础上添加了n-gram编码器，通过融合字符与n-gram表示，有效提升了模型对中文语义结构的理解。实验证明，ZEN在中文分词、命名实体识别、文本分类等多个任务上优于传统方法，表明其多粒度建模策略在中文NLP中具有重要价值。

3 PolicyBERT

3.1 研究方法

本文提出的 PolicyBERT 的模型架构如图 1 所示。

3.1.1 模型输入

令输入的所有政策文件定义为

$$Doc = \{d_1, d_2, \dots, d_{N^{doc}}\} \quad (1)$$

其中 d_i 为第 i 个政策文件, N^{doc} 为政策文件的总数。我们将每个政策文件 d_i 按照标点、换行符等标识分为 N^{d_i} 个句子。

$$d_i = \{s_1, s_2, \dots, s_{N^{d_i}}\} \quad (2)$$

假设每个句子中的词数和字数分别为 k_w 和 k_c , 则每个句子 s_j 可以表示为:

$$\begin{aligned} s_j &= W_j = C_j \\ W_j &= \{w_1, w_2, \dots, w_{k_w}\} \\ C_j &= \{c_1, c_2, \dots, c_{k_c}\} \end{aligned} \quad (3)$$

其中 w_i 为第 i 个词, c_i 为第 i 个字。 k_w 和 k_c 分别为句子中的词和字符的数量。字符数量 k_c 由句子长度决定。为了更好地对应词和字的关系, 我们使用一个匹配矩阵 (Matching Matrix) $\mathcal{M}_j \in N^{k_w^{max} \times k_c}$, 其中每一个元素的取值为

$$m_{pq} = \begin{cases} 1, & \text{if } c_p \in w_q \\ 0, & \text{elif } c_p \notin w_q \end{cases} \quad (4)$$

其中 k_w^{max} 为预先设定的单个句子中词的最大数量, 本文使用 $k_w^{max} = 40$ 。

实际输入时, 模型使用输入的句子 s_j 的字符表示 C_j 与词表 V 进行嵌入。词表 V 由 Hanlp 提供的分词接口预先构建, 共 2662 个词。对于 C_j , 提取其中的所有存在于 V 的词, 记录其索引、起始位置、长度和内容。构建上文提到的匹配矩阵 \mathcal{M}_j , 将所有词的顺序打乱, 以避免模型学习到词表的顺序信息。对于不足 k_w^{max} 的句子, 使用零填充 (Zero Padding) 补齐。如此一来, 构成了长度为 k_w^{max} 词嵌入 C_j 。

3.1.2 多层融合

用于处理字符输入 C_j 和词输入 W_j 的模型架构相似, 均使用了多层的 Transformer 编码器。每一层的输入为上一层的输出, 经过多头自注意力 (Multi-head Self-Attention) 处理后。每一层字符编码器的输出为 $C^{(l)}$, 每一层词编码器的输出为 $W^{(l)}$ 。

为了融合字符和词的表示, 本文使用了两种融合方式: 门控机制 (Gated Mechanism) 和多头注意力机制 (Multi-head Attention Mechanism)。

门控机制通过一个包含全连接层 (Fully Connected Layer) Dense、Sigmoid 激活函数和偏置项的门控单元, 用字符和词的拼接表示, 计算要融合哪些词嵌入的信

息。得到新的字符表示 $C^{(l+1)}$ ：

$$gate^{(l)} = \sigma^{(l)}(Dense(C^{(l)}, W^{(l)})) + b^{(l)} \quad (5)$$

$$C^{(l+1)} = C^{(l)} + gate^{(l)} \cdot W^{(l)} \quad (6)$$

其中每个线性层的偏置（bias）被初始化为常数 $b^{(0)} = 5.0$ ，这样通过 sigmoid 激活函数后，初始门控值接近于 1，这意味着初始状态下模型倾向于直接保留字符表示 $C^{(l)}$ ，与ZEN中直接相加 $C^{(l)}$ 和 $W^{(l)}$ 的原始方法处在同一出发点。

多头注意力机制则是通过计算字符和词的注意力权重，来决定如何融合两者的信息。具体来说，对于每一层的字符表示 $C^{(l)}$ 和词表示 $W^{(l)}$ ，多头注意力机制将字符表示作为查询（Query），词表示作为键（Key）和值（Value），输入到多头注意力模块中。通过计算注意力权重，模型能够捕捉字符和词之间的交互关系，从而生成融合后的表示。

具体实现如下：首先，使用字符表示 $C^{(l)}$ 作为查询 Q ，词表示 $W^{(l)}$ 作为键 K 和值 V ，输入到多头注意力模块中：

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (7)$$

其中 d_k 是键的维度，用于缩放点积以稳定梯度。

多头注意力机制通过多个独立的注意力头来捕捉不同的特征表示：

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (8)$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (9)$$

其中 W_i^Q, W_i^K, W_i^V 是每个注意力头的参数矩阵， W^O 是输出的线性变换矩阵。

在融合过程中，模型还引入了残差连接（Residual Connection）和层归一化（Layer Normalization）以增强稳定性：

$$C^{(l+1)} = \text{LayerNorm}(C^{(l)} + \text{Dropout}(\text{MultiHead}(C^{(l)}, W^{(l)}, W^{(l)}))) \quad (10)$$

模型的最后一层输出为 $C^{(L)}$ ，将被用于下游任务的微调。

3.2 实验准备

3.2.1 数据集

为验证本文提出的融合式中文编码模型在真实领域场景下的有效性，我们构建了一个面向政策文本理解的中文语料数据集。该数据集聚焦于中国语境下的政府文

件与政策文本，具备鲜明的领域特征和丰富的语言结构，是评估模型在政策解读、政务问答等任务中的理想语料基础。

本研究所用政策文本数据集由网络爬虫技术自动采集所得，涵盖了中国大陆多个省市自治区政府官方网站上发布的各类政策文件、政府工作报告、发展规划、法规条例、政策解读书籍摘要等正式文本资料。具体爬取来源包括但不限于各地政府门户网站（如北京、上海、浙江、四川等）、国家发展改革委员会、财政部、教育部等中央部委官网、政策解读平台与政务服务平台发布的权威二次文本。所有文本均为公开发布内容。

最终构建的政策文本语料共计 99 篇文档，涵盖不同政策类别与区域来源，包括宏观经济、财政税收、乡村振兴、教育改革、公共卫生等多个细分领域。文本总句子数为 14,652 个句子，总字符数达约38万汉字，句子平均长度为 26.1字/句，具备典型政策文风特征（长句、嵌套多、名词短语密集）。

3.2.2 数据预处理

在正式输入模型训练与评估之前，需对原始政策文本数据进行系统性预处理，以确保语料干净、结构统一，并充分提取粒度丰富的语言单元以用于编码器输入。由于部分政策文本来自于扫描版文档或PDF转换结果，存在典型的OCR噪声，如断行错误、多余换行符、标点残缺、非正文符号嵌入等。我们对文本进行了如下处理：

将所有非段落边界的换行符（`\n`）替换为空格，避免长句被错误截断；修复常见中英文混排标点错误（如“:”变为“：”），统一中文全角标点风格；移除低频率出现的非Unicode汉字字符、控制符（如`\x0c`）、多余空格或乱码；清洗后文本结构更加规整，显著提升了后续“词级信息”提取的质量与鲁棒性。

政策类文本中，常包含表格信息、超链接、页脚编号、批注引用等非自然语言结构。这些信息对于模型训练而言往往是噪声，需予以剔除。本文采用基于正则表达式的规则方法进行清理，匹配带有明显网格、对齐线性结构的文本片段，如“| xxx | yyy |”或“——”连接的列名，并删除。去除所有形如“http[s]”、“.com”的URL链接；利用常见模式（如“第x页”、“附件x”、“××年××号”）识别页脚、页码信息，并从段落中剥离；去除括号中的编号批注（如“(1)”、“[图1]”），并重构自然语言句式。

相比基于n-gram的策略，HanLP提供了更语义驱动的中文分词工具，其基于神经网络词法分析模型，并支持自定义词典增强分词效果。使用 HanLP提供的api接口对清洗后的句子进行切词，统计词表中的词频，剔除低频词（出现次数小于10次）以

减少噪声对模型训练的影响。最终构建的词表包含2662个词，覆盖了99.8%的句子。

3.2.3 微调数据集构建

为评估改进后的中文文本编码器对上下文建模能力的增强效果，本文构建了一个适用于中文领域政策文本的**Next Sentence Prediction (NSP)**任务数据集。不同于英文NSP任务中基于自然段的句子抽取，中文文本中**句子划分粒度模糊、标点使用灵活**，因此我们结合句号（“。”）与逗号（“，”）进行多粒度分割，设计出更贴合中文语言结构的样本生成策略。

正样本的构建遵循“语义连续、上下文自然”的原则。我们首先使用HanLP对原始政策语料进行句子划分，按“中文句号（。）”作为分句边界，得到句子集合：

$$\mathcal{S} = \{s_1, s_2, \dots, s_n\} \quad (11)$$

对于每个句子我们利用“逗号”划分子句：

$$s_i = \{s'_{i,1}, s'_{i,2}, \dots, s'_{i,k}\} \quad (12)$$

将相邻的子句对 $(s_{i,k}, s_{i,k+1})$ 作为正样本对 (s_i, s_{i+1}) ，即：

$$(s_{i,k}, s_{i,k+1}) \in \mathcal{P} \quad (13)$$

最终得到的正样本对数量为：

$$|\mathcal{P}| = 23374 \quad (14)$$

针对负样本，我们设计了两种生成策略，分别体现不同的训练目标与数据分布假设。

方案一：正负样本比例1:1，包含“句子顺序扰动”样本
该策略下，负样本数与正样本数相等，即：

$$|\mathcal{N}_1| = |\mathcal{P}| = 23374 \quad (15)$$

在构造负样本 $(s'_i, s'_j) \in \mathcal{N}_1$ 时，采用如下两类样本来源：

1. 顺序颠倒型（20%）：从正样本中选取20%的句子对 (s_i, s_{i+1}) ，反转顺序构造 (s_{i+1}, s_i) ；

2. 跨文档随机型（80%）：从整个语料库中随机选取两个不相邻句子 (s'_i, s'_j) ，其中 s'_i 和 s'_j 可来自不同文章或不同段落，确保语义无关。

该策略优点在于正负样本数量平衡，有助于训练模型识别语义连贯性，并关注句子顺序是否合理。我们将该方案的数据集称为PolicySM-1（Policy Sentence Matching 1），并在后续实验中与方案二进行对比。

方案二：正负样本比例1:5，基于篇内语义混淆构造

在第二种策略中，我们设计更具挑战性的负样本：从同一篇文章中抽取相隔2至5句话的句子对，并在其内部使用逗号划分子句以混淆语义边界。即，对于句子序列 $\{s_i\}$ ，构造如下样本对：

$$(s_{i,k}, s_{j,l}) \quad \text{其中} \quad j = i + \delta, \quad \delta \in \{2, 3, 4, 5\} \quad (16)$$

其中 $s_{i,k}$ 和 $s_{j,l}$ 分别为句子 s_i 和 s_j 中用逗号划分出的任一子句。最终采样形成：

$$|\mathcal{N}_2| = 5 \times |\mathcal{P}| = 116870 \quad (17)$$

该方案的设计意图在于强化模型对篇内局部干扰与语义跳跃的辨别能力，提升编码器对更细粒度上下文关系的建模能力。尤其是在RAG任务中，模型需要能够区分相关与不相关的句子对，以便在检索阶段准确匹配问题与答案。而更多的负样本数量也有助于模型学习到更丰富的语义信息，避免过拟合。因此，方案二的负样本构造更贴近实际应用场景，能够有效提升模型的泛化能力。我们将方案二的数据集称为PolicySM-2（Policy Sentence Matching 2）。

我们分别采用方案一与方案二构建了两个NSP训练数据集，分布如表 1 所示。

3.2.4 骨干模型

有关于模型的具体实现，本文基于 *HuggingFace* 的 *Transformers* 库，使用了 *ZEN-pretrain-base* 模型和 *bge-base-zh-v1.5* 模型。前者是基于 BERT 的中文预训练模型，拥有预训练的字符嵌入与词嵌入参数。后者是 BAAI 提出的中文通用嵌入模型，主要用于RAG任务中的文本嵌入生成。我们将 *bge-base-zh-v1.5* 的主体 BERT 部分参数提取出来作为骨干字符嵌入模型，其余的参数则使用 *ZEN-pretrain-base* 模型进行初始化。微调过的 *bge-base-zh-v1.5* 模型将被用于后续的RAG系统中。

数据集	正样本数	负样本数	正负比例	总样本数
PolicySM-1	23374	23374	1:1	46748
PolicySM-2	23374	116870	1:5	140244

表 1: NSP任务样本分布

在后续实验部分，我们将分别基于两种方案训练NSP任务模型，并评估不同样本构造策略对文本编码器语义理解能力的影响。

3.2.5 训练配置

实验在一台高性能GPU服务器上进行，配置如下：

GPU: NVIDIA RTX 4090 \times 1, 显存 24 GB

CPU: 20 核 Intel Xeon 处理器

内存: 80 GB

工作空间: 50 GB SSD 存储用于中间缓存与模型权重

框架: PyTorch 2.0, python 3.8, cuda 11.8

该配置确保了在处理大规模预训练模型和多层融合机制时具备稳定的训练性能和内存调度能力。

3.3 实验结果

3.3.1 PolicySM-1

我们在PolicySM-1数据集上测试了不同融合方法与Hanlp分词的效果。在这部分实验中，我们仅使用ZEN-pretrain-base模型作为基础模型，使用了不同的融合方法（门控机制、多头注意力机制）与原始论文中直接相加进行对比。并对比了使用Hanlp分词与使用n-gram表示的效果。

预训练和下游任务的训练超参数分别如表 2 和表 3 所示。

参数名称	参数取值	描述
epochs	3	总训练轮数
train_batch_size	32	训练批量
learning_rate	3e-5	初始学习率
warmup_proportion	0.9	学习率预热比例

表 2: PolicySM-1预训练超参数

参数名称	参数取值	描述
max_seq_length	128	最大输入长度
train_batch_size	32	训练批量
eval_batch_size	8	评估的批量大小
learning_rate	5e-5	初始学习率
num_train_epochs	3	总训练轮数
warmup_proportion	0.9	学习率预热比例

表 3: PolicySM-1下游任务超参数

实验结果由表 4 所示。可以看到，使用Hanlp分词的模型在所有融合方法中表现最好，达到了0.9132的准确率。使用n-gram表示的模型在Attention-based Fusion下也取得了不错的效果，达到了0.9121的准确率。这意味着，使用Hanlp分词的模型在处理中文政策文本时，能够更好地捕捉到词语之间的关系和上下文信息，从而提高了模型的性能。而两种融合方法与原始论文中直接相加的效果相比，均有了显著提升，说明引入门控机制和多头注意力机制能够有效增强模型对中文政策文本的建模能力。

3.3.2 PolicySM-2

我们在PolicySM-1数据集上同时测试了两个模型（ZEN-pretrain-base和bge-base-zh-v1.5）在不同融合方法下的效果。预训练和下游任务的训练超参数分别如表 5 和表 6 所示。由于先前的实验已经验证了使用Hanlp分词的模型在所有融合方法中表现最好，因此我们在这部分实验中仅使用了Hanlp分词。

融合方法	分词方法	准确率
None	n-gram	0.8867
Gated Fusion	n-gram	0.8959
Attention-based Fusion	n-gram	0.9121
None	Hanlp	0.9035
Gated Fusion	Hanlp	0.8981
Attention-based Fusion	Hanlp	0.9132

表 4: PolicySM-1下游任务结果

参数名称	参数取值	描述
epochs	5	总训练轮数
train_batch_size	32	训练批量
learning_rate	3e-5	初始学习率
warmup_proportion	0.9	学习率预热比例

表 5: PolicySM-2预训练超参数

参数名称	参数取值	描述
max_seq_length	128	最大输入长度
train_batch_size	32	训练批量
eval_batch_size	8	评估的批量大小
learning_rate	5e-5	初始学习率
num_train_epochs	3	总训练轮数
warmup_proportion	0.9	学习率预热比例

表 6: PolicySM-2下游任务超参数

实验结果由表 7 和表 8 所示。我们可以看到，在预训练任务中，使用了门控机制和多头注意力机制的模型在MLM Loss、MLM 准确率和困惑度上均有显著提升，尤其是使用多头注意力机制的模型在所有指标上均达到了最优值。这表明，融合式中文编码模型能够有效地捕捉到字符和词之间的关系，从而提高了模型的性能。

而在下游任务中，使用了门控机制和多头注意力机制的模型在NSP准确率上也

有显著提升，尤其是使用多头注意力机制的模型在NSP准确率上达到了最优值。这表明，融合式中文编码模型能够有效地捕捉到上下文信息，从而提高了模型的性能。

模型	融合方法	MLM Loss	MLM acc	困惑度
BGE	none	1.0151	0.7837	2.76
	gate	0.9763	0.7924	2.65
	attn	0.9218	0.8029	2.51
ZEN	none	0.9064	0.8036	2.48
	gate	0.9332	0.7995	2.54
	attn	0.8952	0.8059	2.45

表 7: PolicySM-2预训练结果

模型	融合方法	NSP acc
BGE	none	0.9236
	gate	0.9311
	attn	0.9272
ZEN	none	0.9260
	gate	0.9325
	attn	0.9274

表 8: PolicySM-2下游任务结果

4 未名问政RAG系统



图 1: 一只雪貂

5 结论

数据科学技术为金融风控提供了强大支持。未来，结合更丰富的数据源与更先进的深度模型，有望进一步提升预测能力。

参考文献

- [1] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.
- [2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, March 2003.
- [3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.
- [4] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.
- [5] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

- [8] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration, 2019.
- [9] Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. Zen: Pre-training chinese text encoder enhanced by n-gram representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4729–4740, Online, November 2020.
- [10] Shizhe Diao, Ruijia Xu, Hongjin Su, Yilei Jiang, Yan Song, and Tong Zhang. Taming pre-trained language models with n-gram representations for low-resource domain adaptation. 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 3336–3349, Online, August 2021. Association for Computational Linguistics.