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

北京大学本科毕业论文导师评阅表

学生姓名		学号	
院系		专业	
指导教师		职称	
毕业论文题目			
导师是否同意参加毕业论文答辩		建议成绩（可选填）	
导师评语	（包括但不限于对论文选题意义、行文逻辑、专业素养、学术规范以及是否符合培养方案目标等方面评价） 导师签名：　　_____ 年 __ 月 __ 日		

摘要

本文提出了一种基于融合式中文编码模型的文本编码器——PolicyBERT，旨在提升中文政策文本的语义理解与生成能力。通过引入 HanLP 分词、门控机制和多头注意力机制，PolicyBERT 能够有效融合字符级和词级信息，增强对中文政策文本的建模能力。本文在多个任务与公开数据集上进行了对比实验，实验结果表明，PolicyBERT 在多个任务上均表现出色，尤其是在中文分词（CWS）和词性标注（POS）任务中，分别达到了 0.9816 和 0.9716 的 F1 值，显著优于其他模型。此外，在句子对匹配（SPM）任务中，PolicyBERT 的多头注意力机制融合方法取得了 0.9016 的最高 F1 值，显示出其在捕捉上下文语义关系方面的优势。本文构建了两个适用于中文政策文本的下一句预测（NSP）任务数据集（PolicySM-1 和 PolicySM-2），并在这些数据集上进行了模型微调。实验结果表明，使用多头注意力机制的模型在 NSP 任务中表现最佳，达到了 0.9325 的准确率。此外，本文基于微调后的模型，使用 Ollama 与 langchain-chatchat 架构搭建了一个基于检索增强生成（RAG）机制的智能问答系统“未名问政”，实现了从文本检索到问答生成的全流程部署。该系统具备部署灵活、响应快速、扩展便捷等特点，适用于政务智能问答、政策知识库构建与自动解读等场景。本文的研究为中文政策文本处理提供了新的方法和工具，具有重要的理论意义和实际应用价值。

关键词：中文政策文本，融合式编码模型，PolicyBERT，分词，检索增强生成，未名问政

Abstract

This paper proposes a text encoder, PolicyBERT, based on a fusion-based Chinese encoding model, aiming to enhance the semantic understanding and generation capabilities of Chinese policy texts. By introducing HanLP word segmentation, gating mechanisms, and multi-head attention mechanisms, PolicyBERT effectively integrates character-level and word-level information, improving its modeling capabilities for Chinese policy texts. Comparative experiments were conducted on multiple tasks and public datasets, and the results demonstrate that PolicyBERT performs exceptionally well across various tasks. Specifically, it achieved F1 scores of 0.9816 and 0.9716 in Chinese Word Segmentation (CWS) and Part-of-Speech Tagging (POS) tasks, respectively, significantly outperforming other models. Additionally, in the Sentence Pair Matching (SPM) task, the multi-head attention mechanism fusion method of PolicyBERT achieved the highest F1 score of 0.9016, showcasing its advantages in capturing contextual semantic relationships. This paper also constructed two datasets for Next Sentence Prediction (NSP) tasks tailored to Chinese policy texts (PolicySM-1 and PolicySM-2) and fine-tuned the model on these datasets. Experimental results show that models using the multi-head attention mechanism performed best in the NSP task, achieving an accuracy of 0.9325. Furthermore, based on the fine-tuned model, this paper developed an intelligent question-answering system, "WeiMing Policy Q&A," using the Ollama and langchain-chatchat architecture, which is based on a Retrieval-Augmented Generation (RAG) mechanism. This system achieves end-to-end deployment from text retrieval to answer generation and features flexible deployment, fast response, and easy scalability, making it suitable for scenarios such as intelligent government Q&A, policy knowledge base construction, and automatic interpretation. This research provides new methods and tools for processing Chinese policy texts, with significant theoretical and practical value.

Keywords: Chinese policy texts, fusion-based encoding model, PolicyBERT, word segmentation, Retrieval-Augmented Generation，WeiMing Policy Q&A

目录

1 引言

1.1 研究背景

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。

在自然语言处理发展早期，文本表示主要依赖于基于词典的特征工程方法，如 TF-IDF ，通过使用词频和逆文档频率来量化词的重要性 [1]。而主题模型（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，提升语义理解能力 [2]。随着深度学习（Deep Learning）的兴起，词向量技术（如 Word2Ve [3] 和 GloVe [4]）通过大规模的无标注语料训练，将词或者字映射为低维实数向量，从而捕捉词间相似性。但其“静态”表示的特点是其难以区分多义词，并且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）为自然语言处理领域带来了革命性的变革。其中最具有代表性的 BERT 首创了“掩码语言模型”和“下游任务微调”范式，通过在大规模语料库上进行预训练并在具体任务上微调，大幅提升性能，从此奠定了通用 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模型在多项任务上的表现 [7]。ERNIE 则融入知识图谱与实体信息，在预训练中引入结构化的外部知识，进一步增强了模型性能 [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由语言差异带来的挑战。中文天然缺乏空格作为词语边界标记。由于模型在对文本进行编码前通常需先进行分词处理（Tokenization）。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，但均有可能在某中语境中存在。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，也即将每个汉字视为独立的一个 token。这样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，以及词本身所包含的语义信息。例如将“数据治理”拆解为“数”、“据”、“治”、“理”，使模型需要通过更复杂的理解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 的具体子领域应用中，领域文本（Domain-specific Text）处理逐渐成为研究热点。尤其是在法律、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的要求。模型不仅要能理解个体 token 的语义，还要具备对领域专有词、实体的整体感知能力。这使得纯粹依赖字粒度编码的 BERT 类模型在此类中文任务上相比英文存在明显缺陷。若模型不能识别和建模“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。

HanLP 是一个开源的自然语言处理工具包，提供了多种中文领域的 NLP 任务工具，如分词、词性标注、命名实体识别等 [11]。其分词器基于循环神经网络（RNN）和 CRF（条件随机场）模型，能够高效且精准地处理中文文本中的分词问题。其提供的 api 接口能够返回一个句子中所有的词，利用这些词可以为模型引入更丰富的词级语义信息。

本文在 ZEN 的基础上，提出了一种基于融合式中文编码模型的文本编码器——PolicyBERT，旨在提升类 BERT 模型对于中文政策文本的语义理解能力。该模型通过使用 HanLP 分词引入了更精准、丰富的词级语义信息，并使用门控与注意力机制提高了模型对词级语义信息的利用能力，增强模型对文本的表示能力。

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）的系统架构在政策文本处理领域展现出前所未有的潜力。RAG 是一种将大型语言模型与外部知识库相结合的创新方法，其核心在于大语言模型在接受了用户的提示词后，先在知识库中进行检索，将结果与原始提示词进行组合并进行回答。研究表明，RAG 在多个评估指标上优于传统的领域特定微调方法，尤其在知识检索任务中表现出更高的准确性和可靠性 [12]。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，这对模型的能力提出了更高的要求。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了更好地适应特定领域的语义需求，提升 RAG 系统性能的关键。

此外，BGE（BAAI General Embedding）模型是由北京智源人工智能研究院（BAAI）开发的一款通用嵌入模型，专为中文自然语言处理任务设计 [13]。作为一款高效的文本嵌入生成工具，BGE 模型在语义一致性和检索精度方面表现卓越。其核心优势在于通过大规模中文语料的预训练，能够生成具有高语义相关性的文本向量，从而在检索增强生成（RAG）任务中展现出强大的知识库检索能力。尤其是 $bge-base-zh-v1.5$ 版本，凭借其轻量化的架构和优化的嵌入生成算法，不仅在中文文本检索任务中实现了更高的精度，还为构建高效的知识检索模块提供了坚实的技术支持。

Ollama 是一个高效的模型部署工具，其支持本地化部署大规模语言模型，能满足离线部署、子领域模型微调等需求。本文中，我们采用 Ollama 实现本地部署 $Qwen2.5:7b$，一款由阿里巴巴达摩院推出的中文大语言模型，其同时具备较小的体积和卓越的文本生成能力、上下文理解能力 [14]。此外，BGE (BAAI General Embedding) 系列模型在 RAG 任务中的知识库检索表现出色，尤其是 $bge-base-zh-v1.5$，其在用于知识检索的中文文本嵌入生成方面具有较高的语义一致性和精度，为构建高效的知识检索模块提供了坚实的基础。

在架构设计上，langchain-chatchat 是一款灵活且高效的框架，专为复杂的 RAG 系统设计，能够无缝集成检索、生成、知识库管理等模块，支持多阶段任务处理 [15]。langchain-chatchat 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 本地部署的 $Qwen2.5:7b$ 和 $bge-base-zh-v1.5$，构建一个高效的 RAG 系统————“未名问政”。该系统将利用 $Qwen2.5:7b$ 进行文本生成，并通过使用 ZEN 架构微调的 $bge-base-zh-v1.5$ 实现高效的政策文本检索。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。

1.2 研究意义

本文提出的 PolicyBERT 架构，结合了 HanLP 中文分词、门控机制和多头注意力机制，通过高效利用词级语义信息，显著提升了中文政策文本的语义理解与生成能力，为中文自然语言处理（NLP）领域开辟了一条全新的研究路径。中文语言因其独特的字符结构和语义复杂性，一直以来在 NLP 领域面临诸多挑战。PolicyBERT 通

过引入融合式编码模型，探索了字符级与词级信息的深度融合方法，为解决中文语言特性带来的建模难题提供了创新性的解决方案。这种方法不仅能够更精准地捕捉中文文本的语义特征，还为其他语言的自然语言处理研究提供了可借鉴的经验，具有重要的理论价值和实践意义。

PolicyBERT 通过结合 HanLP 分词技术，充分挖掘了中文文本中词级语义信息的潜力。传统的中文 NLP 模型多采用字符级编码策略，虽然规避了分词歧义问题，但在捕捉词语整体语义和上下文关系方面存在不足。PolicyBERT 通过引入门控机制和多头注意力机制，进一步增强了模型对词级信息的利用能力，使其能够在复杂的中文政策文本中更准确地建模语义关系。这一创新不仅为中文 NLP 领域提供了新的技术思路，也为其他语言的文本处理提供了启发。

PolicyBERT 的研究为中文政策文本的智能化处理提供了重要支持。政策文本通常具有术语密集、结构复杂、逻辑严谨等特点，对模型的语义理解能力提出了更高要求。PolicyBERT 通过深度融合字符级和词级信息，显著提升了模型对政策文本的理解与生成能力，为政策解读、知识库构建、智能问答等实际应用场景提供了强有力的技术支撑。

此外，本文构建的“未名问政”系统，作为 PolicyBERT 在实际场景中的应用示例，不仅验证了模型在检索增强生成（RAG）任务中的优越性能，还展示了其在政务智能问答、政策知识库构建与自动解读等场景中的巨大潜力。该系统通过将检索与生成相结合，为用户提供了快速、精准的政策信息获取途径，进一步降低了政策解读的门槛，提升了政策传播的效率。这一系统的成功实践，充分证明了 PolicyBERT 在实际应用中的价值，也为未来基于 RAG 架构的智能系统设计提供了宝贵经验。

”未名问政“RAG 系统所具备的灵活部署、快速响应和便捷扩展的特点，使其在政务智能问答、政策知识库构建与自动解读等场景中具有广泛的应用前景。通过结合 Ollama 本地部署的 $Qwen2.5:7b$ 和 $bge-base-zh-v1.5$，该系统实现了从文本检索到问答生成的全流程部署，为用户提供了高效、精准的政策信息获取途径。这一系统不仅为政策文本处理提供了新的解决方案，也为未来基于 RAG 架构的智能系统设计提供了宝贵经验。

1.3 研究内容

本文的研究内容围绕中文政策文本的语义理解与生成展开，旨在通过构建融合式中文编码模型，提升模型对政策文本的建模能力，并验证其在实际应用场景中的有效性。本文提出了一种融合式中文编码模型——PolicyBERT，结合 HanLP 分词技术、

门控机制和多头注意力机制，探索字符级与词级信息的深度融合方法。通过引入门控机制，模型能够动态调节字符与词级信息的权重；通过多头注意力机制，模型能够捕捉更复杂的上下文关系，从而提升对中文政策文本的语义理解能力。

为验证模型的有效性，本文构建了一个面向中文政策文本的语料数据集，涵盖多个领域的政策文件。通过数据清洗、分词与词表构建等预处理步骤，确保数据的高质量与多样性。此外，本文设计了两个适用于下一句预测（NSP）任务的数据集（PolicySM-1 和 PolicySM-2），分别采用不同的负样本生成策略，以评估模型在不同语义关系建模任务中的表现。

本文在公开数据集和自构建数据集上对 PolicyBERT 模型进行训练与评估。通过对比不同融合方法（如门控机制、多头注意力机制）与分词策略（如 n-gram、HanLP 分词），验证模型在中文分词、词性标注、句子对匹配等任务中的性能提升。

基于微调后的 PolicyBERT 模型，本文构建了一个基于检索增强生成（RAG）机制的智能问答系统“未名问政”。系统结合 PolicyBERT 的政策文本嵌入能力与 $Qwen2.5:7b$ 模型的生成能力，实现了从文本检索到问答生成的全流程本地部署，适用于政务智能问答、政策知识库构建等场景。

1.4 研究架构

在本研究中，首先通过 HanLP 分词、门控机制与多头注意力机制等关键技术，搭建融合式中文编码模型——PolicyBERT。其根本思路在于，从字符级与词级这两个层面挖掘文本内在语义，并通过门控单元与多头注意力机制实现多粒度信息的灵活融合，从而应对中文政策文本词界模糊、术语密集等挑战。随后，在模型训练与评估阶段，选取若干公开数据集与自构建数据集，通过与不同分词策略或融合手段进行对比实验，检验 PolicyBERT 在中文分词、词性标注、句子对匹配等任务上的综合表现。最后，将微调后的 PolicyBERT 嵌入到 RAG（检索增强生成）系统“未名问政”中，与 Ollama 本地部署的语言模型相结合，实现从检索到生成的高效政务问答流程。整篇论文的结构大致分为三部分：第一部分聚焦研究背景、文献综述与模型理念；第二部分着重介绍数据准备、模型设计与实验细节；第三部分则阐述 RAG 系统应用、实验结果与未来展望。通过这一整体安排，期望能够系统呈现 PolicyBERT 技术方案的设计初衷、实验验证与实际落地价值。

本文的主要贡献包括以下几个方面：提出 PolicyBERT 架构，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 $ZEN-pretrain-base$ 的词编码预训练参数拼接，使

用 PolicyBERT 架构进行微调，以实现对政策文本的理解；使用 Ollama 本地部署 $Qwen2.5:7b$ 和 $bge-base-zh-v1.5$，并结合 langchain-chatchat 架构构建 RAG 系统，利用微调后的 $bge-base-zh-v1.5$，实现高效的政策文本检索与生成任务。通过这一系统，本文验证了多层融合词级信息对文本理解效果的提升，以及在特定领域微调模型和优化架构设计对 RAG 系统性能的提升。本文用于 PolicyBERT 的训练，未名问政 RAG 系统的搭建的代码和数据公开在PuHT4213/PolicyBert和PuHT4213/WeiMingPolicyRAG上。

2 相关研究

2.1 早期文本表示模型

在自然语言处理（NLP）领域，文本表示模型经历了从基于词典的特征工程方法到基于深度学习的词向量技术的演变。早期的文本表示方法主要依赖于基于词典的特征工程，相关研究如下所示：

TF-IDF（Term Frequency-Inverse Document Frequency）是一种经典的文本表示方法，通过计算词频和逆文档频率来量化词的重要性 [1]。TF-IDF 方法的核心思想是，某个词在一篇文档中出现的频率越高，且在其他文档中出现的频率越低，则该词对该文档的重要性越高。TF-IDF 方法通过将每个词表示为一个权重值，从而实现了文本的向量化表示。这种方法在信息检索和文本分类等任务中被广泛应用，但其静态表示的特点使得难以捕捉多义词和上下文变化。TF-IDF 方法在信息检索和文本分类等任务中被广泛应用，但其静态表示的特点使得难以捕捉多义词和上下文变化。

Word2Vec 是一种由 Mikolov 等人在 2013 年提出的词向量生成方法，其核心思想是通过神经网络模型将词语映射到一个连续的向量空间中，使得语义相似的词在向量空间中距离更近 [3]。Word2Vec 提供了两种主要的训练方法：CBOW（Continuous Bag of Words）和 Skip-Gram。CBOW 模型通过上下文预测目标词，而 Skip-Gram 模型则通过目标词预测上下文。相比传统的词袋模型，Word2Vec 能够捕捉词语之间的语义关系，例如“国王 - 男人 + 女人 ≈ 女王”这样的类比关系。由于其高效性和良好的语义捕捉能力，Word2Vec 在自然语言处理领域得到了广泛应用。

GloVe（Global Vectors for Word Representation）是 Pennington 等人在 2014 年提出的一种词向量生成方法，与 Word2Vec 不同，GloVe 通过全局统计信息来学习词向量 [4]。具体来说，GloVe 利用词与词之间的共现矩阵，通过构建一个目标函数，使得词向量能够保留词语共现概率的比例关系，从而捕捉词语的语义信息。GloVe 的优

势在于它结合了全局统计信息和局部上下文信息，能够更好地捕捉稀疏数据中的语义关系。由于其理论基础扎实且效果显著，GloVe 同样被广泛应用于各种自然语言处理任务中，如文本分类、情感分析和机器翻译等。

FastText 是 Facebook AI Research 提出的一个词向量生成方法，其核心思想是将词语表示为字符 n-gram 的组合，从而能够捕捉到词语的形态学信息 [16]。FastText 在训练过程中，将每个词拆分为多个 n-gram 片段，并通过这些片段来学习词向量。这种方法使得 FastText 能够处理未登录词（Out-of-Vocabulary Words），即在训练数据中未出现过的词。此外，FastText 还能够捕捉到词语的形态变化，例如单复数、时态等。这使得 FastText 在处理形态丰富的语言时表现出色，尤其是在低资源语言和领域特定任务中。

然而，早期的文本表示模型在处理多义词、上下文依赖性和长距离依赖等问题时存在一定的局限性。为了解决这些问题，研究者们开始探索基于深度学习的上下文敏感模型。

2.2 基于深度学习的文本表示模型

随着深度学习技术的发展，基于神经网络的文本表示模型逐渐成为主流。这些模型通过学习上下文信息，能够更好地捕捉词语的语义关系。

ELMo（Embeddings from Language Models）开创了深度上下文感知词表示的先河 [17]。ELMo 采用多层双向长短期记忆网络（BiLSTM）架构，将字符级输入映射为词级向量表示，然后分别通过前向和后向两层 LSTM 以捕捉上下文信息，最终将所有层的输出按任务需求加权组合，生成每个词在特定上下文中的动态表示。在预训练阶段，ELMo 的前向 LSTM 负责预测下一个词，后向 LSTM 则预测上一个词；预训练完成后，仅需对输出层的组合权重进行微调即可应用于下游任务。得益于这种深度、双向且上下文敏感的表示方式，ELMo 在共指消解、命名实体识别等多种任务上均实现了显著提升，标志着 NLP 从静态词向量走向动态预训练模型的关键转折。

BERT（Bidirectional Encoder Representations from Transformers）则以 Transformer 编码器为基础，进一步革新了预训练范式 [6]。BERT 同时引入“掩码语言模型”（Masked Language Modeling，MLM）与“下一句预测”（Next Sentence Prediction，NSP）两大预训练任务：在 MLM 中，随机掩盖输入序列中 15% 的词并训练模型根据双向上下文恢复它们；在 NSP 中，训练模型判断两句话是否在原文中相邻。BERT 的输入由词嵌入、位置嵌入和句子嵌入三部分组成，其 Base 版拥有 12 层编码器、24 个注意力头；Large 版则扩展至 24 层。大规模预训练后，BERT 在问答、文本分类、

自然语言推断等多项基准测试上刷新了性能记录，成为众多后续模型的基石和标杆。

在此基础上，RoBERTa（A Robustly Optimized BERT Pretraining Approach）对 BERT 的预训练细节进行了系统优化 [7]。RoBERTa 取消了 NSP 任务，采用动态掩码（dynamic masking）、更大规模的批量训练与更多语料（包含 CC-News、OpenWebText、Stories 等），并延长预训练轮次。实验表明，适当调整批量大小、学习率与掩码策略后，RoBERTa 在 GLUE、RACE、SQuAD 等任务上全面超越原始 BERT，凸显了超大规模训练与训练细节对性能的关键影响。

在中文领域，多个团队也对 BERT 进行了适配与改进，以应对中文特有的分词与语义挑战。最具代表性的是“全词 Masking”策略下的 Chinese-BERT-wwm，由哈工大与爱科团队基于中文维基百科和 LTP 分词工具训练，保证整词遮蔽时所有子字同时被掩盖，提升了预训练与下游任务的一致性与表现 [18]。其后出现的 MacBERT 在 RoBERTa 基础上提出“MLM as Correction”策略，即用相似词替换原词而非单一 $[MASK]$ ，并引入 N-gram 遮蔽与句序预测（SOP），在多项中文 NLP 任务上取得了当时 SOTA 成绩 [19]。

百度提出的 ERNIE（Enhanced Representation through kNowledge Integration）模型在 BERT 架构的基础上，引入了实体级和短语级的知识掩蔽策略（entity-level & phrase-level masking），显著扩展了原有掩蔽语言模型的表达能力。该模型通过在预训练阶段融合外部知识图谱中的实体与短语边界信息，使模型不仅能够学习词语的语法关系，还能捕捉更丰富的语义层次，从而在自然语言推理、命名实体识别以及问答系统等任务上取得了显著提升 [8]。这种有监督感知的预训练策略，代表了语言模型从“无知识”向“弱知识引导”迈出的重要一步。

在 ERNIE 的基础上，ERNIE-Gram 进一步加强了模型对短语结构的建模能力。该模型通过显式引入 n-gram 掩蔽机制，使得语言模型可以更有效地捕捉连续词组所蕴含的语义单位。这一策略不仅增强了模型对短语级依赖关系的建模能力，也解决了传统 BERT 中对词语孤立建模的问题。实验表明，ERNIE-Gram 在多个中文自然语言处理任务中，如阅读理解与文本匹配等，均超越了同期的主流模型 [20]，显示出短语建模在中文语境中的强大潜力。

MigBERT 是近年来在中文语料上表现突出的另一创新模型。它的主要特点在于融合了字符级与词级的预训练粒度，通过引入混合粒度建模（character-word joint modeling），使得模型既能保留字符级的精细语义信息，又不失词级的语义整体性。这种设计非常适合中文这种语言单位模糊、缺乏天然词界的特点。通过在预训练阶段引导模型同时关注微观和宏观语言结构，MigBERT 有效提升了模型在命名实体识

别、情感分析等任务中的表现，展示出中文预训练语言模型向多粒度方向演进的趋势 [21]。

从 ELMo 的 BiLSTM 转向 BERT 系列的 Transformer 编码器，再到 RoBERTa 的优化和各类中文预训练变体，多种基于神经网络的文本表示模型持续推动了 NLP 在多语言、多场景下的理解与生成能力。

2.3 子领域微调模型

在自然语言处理（NLP）领域，子领域微调模型的研究逐渐成为一个重要的方向。随着预训练语言模型（PLM）的广泛应用，研究者们开始探索如何在特定领域内对这些模型进行微调，以提升其在特定任务上的性能。子领域微调模型的核心思想是通过在大规模通用语料上进行预训练，然后在特定领域的语料上进行微调，从而使模型能够更好地适应特定领域的语言特点和任务需求。

在医疗领域，BioBERT 是一个专门针对生物医学文本的预训练语言模型 [22]。该模型是在原始 BERT 模型的基础上，使用 PubMed 摘要和 PMC 全文等大规模生物医学文献进行进一步预训练而成。通过这种方式，模型能够更好地学习生物医学语料中的术语表达、上下文特征及专业结构化语言，从而弥补通用语言模型在领域适应性上的不足。BioBERT 在多个生物医学自然语言处理任务中进行了微调，如命名实体识别（NER）、关系抽取（RE）和问答系统（QA）等。实验结果表明，相较于未在生物医学领域预训练的模型，BioBERT 在所有任务中均实现了显著的性能提升，有效验证了在特定领域采用子领域微调模型的重要性和实用性。

SMedBERT 是一种结合结构化语义知识的中文医学预训练语言模型，专为医学文本挖掘与理解任务设计。不同于传统仅基于大规模文本语料进行训练的方式，SMedBERT 在预训练阶段引入了医学知识图谱中的实体邻接信息，构建了实体与其语义关联实体之间的上下文关系。这种设计不仅提升了模型对复杂医学术语的识别能力，也加强了对医学实体间关系的建模效果 [23]。在中文医学命名实体识别、医学关系分类及临床术语归一化等多项任务上，SMedBERT 均表现出优异的性能，显著优于通用中文 BERT 模型和其他医学领域基线模型，展现出知识增强型预训练模型在医学领域中的广阔应用前景。

在医学影像报告生成方面，Medical-VLBERT 模型是一种融合视觉和语言信息的多模态预训练语言模型，特别用于生成 COVID-19 相关 CT 报告 [24]。该模型通过视觉-语言双通道的设计，能够同时接收 CT 影像信息和对应的文本描述，通过交替学习策略，在大规模医学文本语料上进行知识预训练，再将所学语言表示迁移至医学图

像任务中。这种方法不仅提升了模型对医学影像语义信息的捕捉能力，也使其在自动生成结构化医学影像报告方面取得了良好效果。实验表明，Medical-VLBERT 在多个生成质量指标上超越了传统报告生成模型，推动了智能医学影像解读技术的发展。

在法律领域，Legal-BERT 是一个专门针对法律文本设计的预训练语言模型 [25]。它在欧盟和美国等司法管辖区的大规模法律文本数据集（如法条、法院判决、律师备忘录等）上进行预训练，并在多个法律自然语言处理任务中进行微调，如法律文本分类、法律问答系统（Legal QA）以及法律信息检索等任务。该模型充分考虑了法律语言的特殊性——包括其专业术语、高度格式化的结构以及复杂的上下文依赖关系，从而显著提升了在法律类任务中的表现。研究表明，Legal-BERT 在多个主流法律任务数据集上相较于通用 BERT 模型取得了明显的性能提升，进一步证实了领域定制化预训练在高专业语域中的应用价值。

Lawformer 是一种基于 Longformer 架构设计的中文法律语言预训练模型，旨在处理法律文书这类超长文本 [26]。传统 BERT 模型受限于固定长度（如 512 token）的输入窗口，在处理动辄上千字的法律判决文书时往往出现截断信息、语义丢失等问题。Lawformer 引入滑动窗口式的注意力机制，使模型能够在处理长文本时保留全局语义信息。在多个中文法律任务中，Lawformer 表现出色，涵盖了判决预测、法律条款匹配、相似案例检索、法律阅读理解和法律问答等任务，均优于通用预训练模型及其他同类模型。这表明，专为长文档优化的模型结构在法律文本处理中的必要性与有效性。

此外，清华大学人工智能研究院与幂律智能合作，推出了民事文书 BERT 与刑事文书 BERT 两个专门面向中文法律子领域的预训练语言模型 [27]。这两个模型分别基于大规模民事案件判决书和刑事案件裁判文书语料进行训练，充分学习了各自领域特有的法律术语、语法模式及判决逻辑。在案由识别、裁判结果预测、相似案例检索等任务中，两个模型均表现出色，显著优于通用中文 BERT 及其他基础法律模型。此举展示了细粒度领域建模的潜力，也为构建更专业、更精准的法律人工智能系统打下了坚实的基础。

在政策文本处理方面，研究者们也开始探索将 BERT 模型应用于政府政策文档的自动分析与理解任务。例如，有研究尝试将 BERT 模型用于政策文本的主题分类、摘要生成、政策条文匹配等任务，并通过微调提升模型对政策语言中复杂表述与行政术语的理解能力 [28]。相比通用语料，政策文本往往具有高度正式性、政策倾向性及隐含逻辑性，因此对模型语言理解能力提出了更高要求。初步实验结果显示，BERT 在经过政策文本微调后，能够较好地捕捉政策语言的结构与语义特征。然而，目前政

策领域的预训练与微调模型仍处于起步阶段，缺乏大规模高质量的政策语料与专用评估基准，仍需更多系统化的研究与跨领域合作，以推动政务文本智能理解技术的发展。

2.4 词级信息融合模型

在中文自然语言处理（NLP）领域，传统的预训练语言模型（PLM）通常采用字符级建模策略，即将每个汉字视为独立的 token。近年来，部分研究者认识到词级信息在中文文本处理中的重要性，并提出了一系列基于词级信息建模的预训练模型。

ZEN（Pre-training Chinese Text Encoder Enhanced by N-gram Representations）提出了一种专为中文语言特性打造的预训练语言模型，其核心目标在于解决中文文本处理中因分词不确定性而导致的语义模糊问题 [9]。与传统 BERT 模型将每个汉字视为独立 token 的方式不同，ZEN 通过引入 n-gram 表示，显式建模多粒度的词片段信息，从而在捕捉词级语义的同时，兼顾了字符级的表达能力。在架构设计上，ZEN 在字符级 BERT 的基础上创新性地添加了 n-gram 编码器，通过融合字符与 n-gram 表示的多层次信息，显著增强了模型对中文复杂语义结构的理解能力。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，充分展现了其多粒度建模策略在中文 NLP 领域的重要价值。

PTWA（Pre-Training with Word Attention）则针对中文命名实体识别（NER）任务提出了一种全新的预训练方法，其核心在于通过引入词注意力机制，进一步强化字符表示与词级语义之间的交互，从而实现对实体边界和类别的更精准识别 [29]。在实验中，PTWA 在多个中文 NER 基准数据集上表现出色，显著优于传统的字符级预训练模型，验证了其在中文 NER 任务中的有效性和先进性。

Wenbiao Li 等人（2022）在其研究《Exploiting Word Semantics to Enrich Character Representations of Chinese Pre-trained Models》中，针对中文预训练语言模型普遍采用字符级建模而忽略词级语义的问题，提出了一种名为“HRMF”（Hidden Representation Mix and Fusion）的新方法 [30]。该方法通过将词嵌入按相似度权重投射到其内部字符嵌入中，增强了字符表示的语义信息，同时利用字符间的混合机制强化词边界信息，并引入词-字符对齐注意力机制，突出重要字符，抑制无关字符的影响。此外，为减轻分词错误传播的问题，作者设计了融合多种分词器结果的集成策略。实验结果显示，该方法在情感分类、句子对匹配、自然语言推理和机器阅读理解等任务中，显著优于基础的中文预训练模型如 BERT、BERT-wwm 和 ERNIE，充分验证了其在丰富字符表示和提升模型性能方面的有效性。

Qiang He 等人（2023）在其论文《Prompt-Based Word-Level Information Injection BERT for Chinese Named Entity Recognition》中，提出了一种名为 PWII-BERT 的模型，旨在通过引入提示（prompt）机制，将词级信息注入预训练语言模型中，从而更有效地捕捉字符与词之间的关联，提升中文命名实体识别（NER）的性能 [31]。具体而言，PWII-BERT 设计了一个词级信息注入适配器（WIIA），利用双仿射机制融合字符和词的特征，并在 Transformer 层中引入类别提示，指导模型关注特定类别的实体信息。实验结果显示，PWII-BERT 在四个中文 NER 基准数据集上均优于现有的主流模型，验证了融合类别信息和词汇特征对于提升中文 NER 性能的显著作用，同时也为中文自然语言处理领域提供了新的研究思路。

2.5 大语言模型

大语言模型（Large Language Model, LLM）是近年来自然语言处理领域的一个重要研究方向。它们通常基于深度学习技术，尤其是 Transformer 架构，通过在大规模文本数据上进行预训练，能够生成高质量的自然语言文本，并在多种下游任务中表现出色。

GPT-4 是 OpenAI 推出的多模态大型语言模型，支持文本和图像输入，具备强大的推理、数学、编程和多语言能力 [32]。该模型在多个基准测试中表现优异，例如在 MMLU（Massive Multitask Language Understanding）基准中取得了 61.8 的分数，领先于其他同类模型。

Qwen2.5 是阿里巴巴推出的中文大语言模型，支持多达 29 种语言，具备强大的知识储备和推理能力 [33]。该模型在多个中文自然语言处理任务中表现出色，尤其在问答、文本生成和对话系统等领域展现了强大的能力。该模型有多个版本，Qwen-7B、Qwen-13B 和 Qwen-70B，分别对应 7 亿、13 亿和 70 亿参数量。Qwen2.5 在多个中文自然语言处理任务中表现出色，尤其在问答、文本生成和对话系统等领域展现了强大的能力。

大语言模型的发展为自然语言处理领域带来了新的机遇和挑战。它们在文本生成、对话系统、机器翻译等任务中展现了强大的能力，推动了人工智能技术的进步。

2.6 基于检索增强生成的领域知识问答系统

RAG（Retrieval-Augmented Generation）是由 Facebook AI（现 Meta AI）在 2020 年提出的一种结合检索与生成能力的预训练语言模型框架，首次发表于《NeurIPS 2020》。RAG 模型通过将生成型语言模型（如 BART）与密集文档检索器（如 DPR）

集成，使得在生成答案时能够动态引用来自外部知识库的文本，从而缓解了传统语言模型面临的“知识封闭”和“幻觉生成”问题。实验表明，RAG 在开放领域问答任务（如 Natural Questions 和 WebQuestions）上显著优于单一生成或检索系统，奠定了“检索增强生成”技术的理论与技术基础。

MedRAG 是由 Zhao 等人于 2025 年提出的医疗领域 RAG 系统，旨在通过引入知识图谱（Knowledge Graph, KG）增强诊断推理能力 [34]。该系统构建了一个四层次的诊断知识图谱，涵盖了各种疾病的关键诊断差异，并将其与电子健康记录（EHR）数据库中的相似病例动态整合。通过在大型语言模型中进行推理，MedRAG 能够提供更准确和具体的诊断建议，并主动提出后续问题以增强个性化医疗决策。实验结果表明，MedRAG 在减少误诊率方面优于现有的 RAG 方法，展示了其在医疗辅助诊断中的潜力。

LegalRAG 是由 Kabir 等人于 2025 年开发的法律领域 RAG 系统，专注于孟加拉国警察公报等双语（英语和孟加拉语）法律文件的问答任务 [35]。该系统结合了现代 RAG 管道和先进的检索方法，提升了信息检索和响应生成的性能。实验评估显示，LegalRAG 在多项评估指标上均优于现有方法，显著提高了法律信息的可访问性和检索效率，特别是在处理政府法律通知等特定任务中表现出色。

FinSage 是由 Wang 等人于 2025 年提出的金融领域 RAG 系统，旨在应对金融文件工作流程中复杂的合规性要求 [36]。该系统引入了多模态预处理管道，统一了多样的数据格式，并生成了块级元数据摘要；采用了增强的稀疏-密集检索系统，结合了查询扩展和元数据感知的语义搜索；并通过直接偏好优化（DPO）微调的领域专用重排序模块，优先考虑合规性关键内容。实验结果显示，FinSage 在由专家策划的 75 个问题上实现了 92.51% 的召回率，在 FinanceBench 问答数据集上的准确率比最佳基线方法高出 24.06%，并已成功部署为在线会议中的金融问答代理，服务超过 1,200 人。

这些系统在各自领域中展示了 RAG 方法的强大能力，证明了检索与生成结合的潜力。中文政策文本处理领域也可以借鉴这些方法，通过结合检索和生成的能力，提升政策文本的理解和生成效果。

2.7 基于文本表示模型的文档检索模型

文本表示模型（Text Embedding Models）在 RAG 系统中起至关重要的作用。北京智源人工智能研究院（BAAI）推出的 BGE 系列模型，特别是其最新成员 BGE-M3，已成为业界关注的焦点。

BGE（BAAI General Embedding）系列模型旨在将文本转换为低维稠密向量，以便进行高效的计算和分析 [13]。初代 BGE 模型基于 BERT 架构，支持包括英语和中文在内的多种语言，广泛应用于检索、重排、聚类和分类等任务。在训练过程中，BGE 模型采用了对比学习和多任务微调策略，确保了其强大的文本处理能力和语义表征能力。

BGE-M3 是 BGE 系列的最新成员，具有多语言（Multi-Linguality）、多功能（Multi-Functionality）和多粒度（Multi-Granularity）等特点 [37]。该模型支持超过 100 种语言，能够处理从短句到长达 8192 个 token 的文档，满足不同粒度的文本表示需求。在检索功能方面，BGE-M3 集成了稠密检索（Dense Retrieval）、稀疏检索（Sparse Retrieval）和多向量检索（Multi-Vector Retrieval）三种方式，为现实世界中的信息检索应用提供了统一的模型基础。

由于 BGE 模型基于 BERT 架构，因此具有较强的可塑性和适应性。研究者们可以根据具体任务的需求，对 BGE 模型进行微调和优化，以提升其在特定领域的表现。

2.8 小结

3 PolicyBERT

3.1 研究方法

3.1.1 模型输入

本文所提出的 PolicyBERT 模型架构如图1所示。其中模型的输入部分由图中左侧所示。

我们将输入模型的所有政策文档定义为

$$Doc = \{d_1, d_2, \ldots, d_{N^{doc}}\} \tag{1}$$

其中 d_i 为第 i 个政策文档，N^{doc} 为政策文档的总数。我们将每个政策文档 d_i 通过基于规则的方式，按照标点、换行符等标识符分为 N^{d_i} 个句子。

$$d_i = \{s_1, s_2, \ldots, s_{N^{d_i}}\} \tag{2}$$

假设每个句子中的词数和字符数分别为 k_w 和 k_c（暂时不考虑如 [UNK]、[CLS]、

[SEP] 等特殊符号的影响），则每个句子 s_j 可以表示为：

$$\begin{aligned} s_j &= W_j = C_j \\ W_j &= \{w_1, w_2, \ldots, w_{k_w}\} \\ C_j &= \{c_1, c_2, \ldots, c_{k_c}\} \end{aligned} \tag{3}$$

其中 w_i 为句子中第 i 个词，c_i 为第 i 个字符。k_w 和 k_c 分别为句子中的词和字符的总数量。字符数量 k_c 直接由句子长度决定。具体对于下游 NSP（Next Sentence Prediction）任务，模型输入 s_j 还可以表示为：

$$s_j = [CLS] + s_j{}^{(1)} + [SEP] + s_j{}^{(2)} + [SEP] \tag{4}$$

其中 $[CLS]$ 和 $[SEP]$ 分别为句子对的起始和分隔符。$s_j^{(1)}$ 和 $s_j^{(2)}$ 分别为句子对中的前句和后句。

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）$\mathcal{M}_j \in N^{k_w \times k_c}$。$\mathcal{M}_j$ 是一个 $k_w \times k_c$ 的 0-1 矩阵，表示句子中的每一个字是否属于某个词。矩阵的每一行对应一个词 w_q，每一列对应一个字 c_p。如果字 c_p 属于词 w_q，则矩阵中对应位置的值为 1，否则为 0。具体来说，$\mathcal{M}_j$ 的每一个元素的取值为：

$$m_{pq} = \begin{cases} 1, & \text{if } c_p \in w_q \\ 0, & \text{elif } c_p \notin w_q \end{cases} \tag{5}$$

实际输入模型时，使用输入句子 s_j 的字符表示 C_j 与词表 V 构建匹配矩阵 $\mathcal{M}_j$ 和词表示 W_j。词表 V 由 Hanlp 提供的分词接口预先构建，并基于规则进行清洗，共包含 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。使用词在词表中的 id，可以构建词表示 W_j：

$$W_j = \{wid_1, wid_2, \ldots, wid_{k_w}\} \tag{6}$$

其中 wid_i 为第 i 个词在词表 V 中的索引。

同时使用公式5构建上文提到的匹配矩阵 $\mathcal{M}_j$，最后将所有词的顺序打乱，以避免模型学习到词表的顺序信息。每句话的最大词数和最大字符数由超参数 ${k_w}^{max}$ 和 k_c^{max} 决定。对于超过的句子，使用截断（Truncation）保留前 ${k_w}^{max}$ 个词和 ${k_c}^{max}$ 个字符。对于不足的句子，使用零填充（Zero Padding）补齐。如此一来，构成了长度为 ${k_c}^{max}$ 的词嵌入 W_j 和长度为 ${k_c}^{max}$ 的字嵌入 C_j。本文设定超参数 ${k_w}^{max} = 40$；而 ${k_c}^{max}$ 为预先设定的单个句子中字符的最大数量，本文在下游 NSP 任务中使用 ${k_c}^{max} = 128$，在公开数据集上的其他任务上使用 ${k_c}^{max} = 128 or 256$。

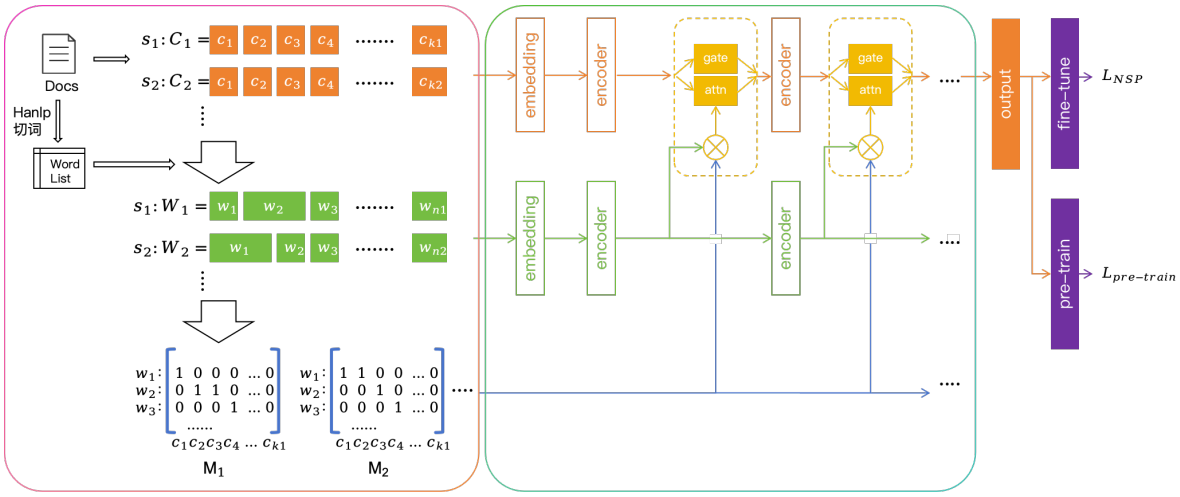

图 1: PolicyBERT 模型架构。左侧为模型的输入部分，包括字符表示 C_j、词表示 W_j 和匹配矩阵 $\mathcal{M}_j$；右侧为模型的主体部分，包括嵌入层、编码层和融合层。其中融合层可以在门控机制和多头注意力机制之间进行切换。

由此，我们构建好了模型的全部输入，包括字符表示 C_j、词表示 W_j 和匹配矩阵 $\mathcal{M}_j$。模型的输入为：

$$X_j = \{C_j, W_j, \mathcal{M}_j\} \tag{7}$$

3.1.2 嵌入、编码与融合

模型的主体部分由图1右侧所示。

模型的输入 X_j 由字符表示 C_j、词表示 W_j 和匹配矩阵 $\mathcal{M}_j$ 组成。为了将这些输入转换为模型可以处理的向量表示，我们使用了嵌入层（Embedding Layer）、编码层（Encoder）和融合层（Fusion Layer）。

字符表示 C_j 和词表示 W_j 首先通过嵌入层进行转换，得到对应的嵌入向量 $C^{(0)}$ 和 $W^{(0)}$：

$$\begin{aligned} C^{(0)} &= text\_embedding(C_j) \\ W^{(0)} &= word\_embedding(W_j) \end{aligned} \tag{8}$$

用于处理字符输入 C_j 和词输入 W_j 的模型架构相似，均为标准的 BERT 架构，使用了多层的 Transformer 编码器。每一层的输入 $C^{(l)}$ 和 $W^{(l)}$，经过多头自注意力（Multi-head Self-Attention）处理后，再经过前馈神经网络进行非线性变换。可以表

示为：

$$\begin{aligned} A^{(l)} &= \text{Attention}(C^{(l)}, \text{mask}) \\ I^{(l)} &= \text{Intermediate}(A^{(l)}) \\ O^{(l)} &= \text{Output}(I^{(l)}, A^{(l)}) \\ C_1^{(l)} &= O^{(l)} \end{aligned} \tag{9}$$

对 $W^{(l)}$ 的处理方式类似，只是对于词表示的中间结果 $W_1^{(l)}$，会令其与匹配矩阵 $\mathcal{M}_j$ 进行相乘，得到词表示中间结果 $W_2^{(l)}$，而对 $C_1^{(l)}$ 则不进行处理。

$$\begin{aligned} W_2^{(l)} &= W_1^{(l)} \cdot \mathcal{M}_j \\ C_2^{(l)} &= C_1^{(l)} \end{aligned} \tag{10}$$

本文在此基础之上，增加了一个融合层（Fusion Layer），用于将字符和词的表示进行融合。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：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 $C^{(l+1)}$：

$$\begin{aligned} gate^{(l)} &= \sigma^{(l)}(Dense(C_2^{(l)}, W_2^{(l)})) + b^{(l)} \\ C^{(l+1)} &= C_2^{(l)} + gate^{(l)} \cdot W_2^{(l)} \end{aligned} \tag{11}$$

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 $C_2^{(l)}$ 和 $W_2^{(l)}$ 的原始方法处在同一出发点。

多头注意力机制：多头注意力机制则是通过计算字符和词向量的注意力权重，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 $C_2^{(l)}$ 和词表示 $W_2^{(l)}$，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。

具体实现如下：首先，使用字符表示 $C_2^{(l)}$ 作为查询 Q，词表示 $W_2^{(l)}$ 作为键 K 和值 V，输入到多头注意力模块中：

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{12}$$

其中 d_k 是键的维度，用于缩放点积以稳定梯度。

多头注意力模块通过多个独立的注意力头来捕捉不同的特征表示：

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \\ \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{13}$$

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：

$$C^{(l+1)} = \text{LayerNorm}(C_2^{(l)} + \text{Dropout}(\text{MultiHead}(C_2^{(l)}, W_2^{(l)}, W_2^{(l)}))) \tag{14}$$

模型的最后一层输出 $C^{(L)}$，将被用于预训练或下游任务的微调。

3.1.3 预训练头

在预训练阶段，模型的最后一层输出 $C^{(L)}$ 被用于两个主要任务：掩码语言模型任务（Masked Language Model, MLM）和下一句预测任务（Next Sentence Prediction, NSP）。

对于掩码语言模型任务，模型首先对输入序列中的部分 token 进行随机掩码处理（即用特殊标记 [MASK] 替换），然后利用 $C^{(L)}$ 预测这些被掩码的 token。具体来说，$C^{(L)}$ 会被输入到一个语言模型预测头（Language Model Prediction Head）中，该预测头由一个全连接层和 softmax 激活函数组成，用于输出每个被掩码 token 的概率分布。预测的概率分布可以表示为：

$$P(\hat{y}_i | C^{(L)}) = \text{softmax}(\text{Linear}(C_i^{(L)})) \tag{15}$$

其中，$C_i^{(L)}$ 表示第 i 个 token 的隐藏状态，$\hat{y}_i$ 表示模型预测的第 i 个 token 的概率分布。

为了计算 MLM 任务的损失，采用交叉熵损失函数（CrossEntropyLoss），公式如下：

$$\mathcal{L}_{MLM} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{16}$$

其中，N 表示被掩码的 token 数量，y_i 表示第 i 个 token 的真实标签，$\hat{y}_i$ 表示模型预测的概率分布。

对于下一句预测任务，模型会将 $C^{(L)}$ 的池化输出（pooled output）输入到一个线性分类器中，用于预测前后句是否具有连续关系。具体来说，池化操作会提取 $C^{(L)}$

中 [CLS] 标记对应的隐藏状态，表示为 $C^{(L)}_{[\text{CLS}]}$，然后通过一个线性层映射到二分类的概率空间：

$$P(\text{NSP}|C^{(L)}) = \text{softmax}(\text{Linear}(C^{(L)}_{[\text{CLS}]})) \tag{17}$$

NSP 任务的损失同样采用交叉熵损失函数，公式如下：

$$\mathcal{L}_{NSP} = -\frac{1}{M}\sum_{j=1}^{M} y_j \log(\hat{y}_j) \tag{18}$$

其中，M 表示样本数量，y_j 表示第 j 个样本的真实标签，$\hat{y}_j$ 表示模型预测的概率分布。

最终，预训练阶段的总损失由 MLM 和 NSP 两部分损失一比一相加求和得到：

$$\mathcal{L}_{\text{pretrain}} = \mathcal{L}_{MLM} + \mathcal{L}_{NSP} \tag{19}$$

通过上述预训练任务，模型能够学习到丰富的上下文语义信息和句子间的逻辑关系，为下游任务的微调提供强大的语义表示能力。

3.1.4 下游任务头

对于下游 NSP 任务，我们使用了一个简单的 Dropout 层和一个线性层来进行分类。Dropout 层用于防止过拟合，线性层用于将模型的输出映射到二分类的概率空间。具体来说，NSP 任务的输出可以表示为：

$$\begin{aligned} NSP^{(L)} &= \text{Dropout}(C^{(L)}) \\ NSP_{output} &= \text{Linear}(NSP^{(L)}) \end{aligned} \tag{20}$$

其中 NSP_{output} 是模型在 NSP 任务上的输出，$C^{(L)}$ 是模型最后一层的输出。

为了计算 NSP 任务的损失（Loss），我们采用了交叉熵损失函数（CrossEntropyLoss）。该损失函数用于衡量模型预测的类别分布与真实标签分布之间的差异。具体来说，假设真实标签为 $labels$，则损失的计算公式为：

$$\mathcal{L}_{NSP} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{2} y_{ij} \log(\hat{y}_{ij}) \tag{21}$$

其中：N 表示样本数量；y_{ij} 表示第 i 个样本的真实标签（one-hot 编码）；$\hat{y}_{ij}$ 表示模型预测的第 i 个样本属于类别 j 的概率；log 表示自然对数。

最终，损失 $\mathcal{L}_{NSP}$ 将用于反向传播，以优化模型参数，提高模型在 NSP 任务上的性能。

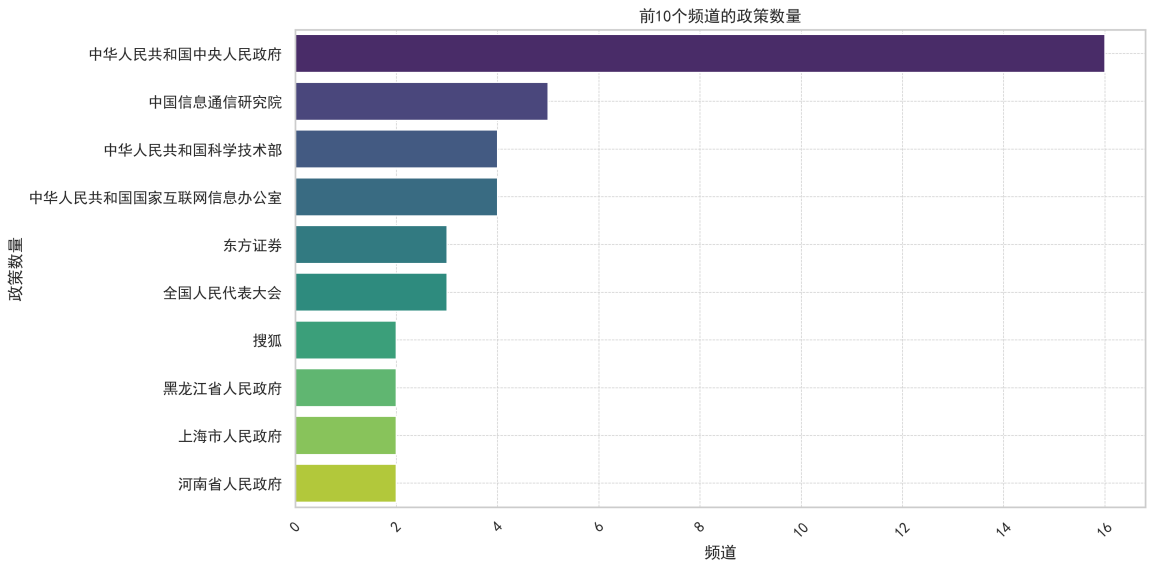

图 2: 政策文本数据集的前十个来源

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，人工整理所得，涵盖了中国大陆多个省市自治区政府官方网站上发布的有关数据治理、信息工程的各类政策文件、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等总计 61 个来源。所有文本均为公开发布内容。数据集中数量排名前十的来源如图2所示。数据集的来源主要集中在中央人民政府、信息通信研究院、科学技术部等。

本政策文本数据集的发布时间与内容长度分布如图3所示。数据集的时间范围为 2010 年到 2025 年。大部分政策集中在 2020 年之后，尤其是 2021 年到 2023 年之间。内容长度主要集中在 10^3 到 10^4 之间。少量政策的内容长度较短或较长。在 2010 年和 2011 年有少量早期政策, 某些年份（如 2014 年到 2018 年）几乎没有政策记录，可能是数据缺失或政策发布较少。随着时间推移，政策的数量逐渐增加。

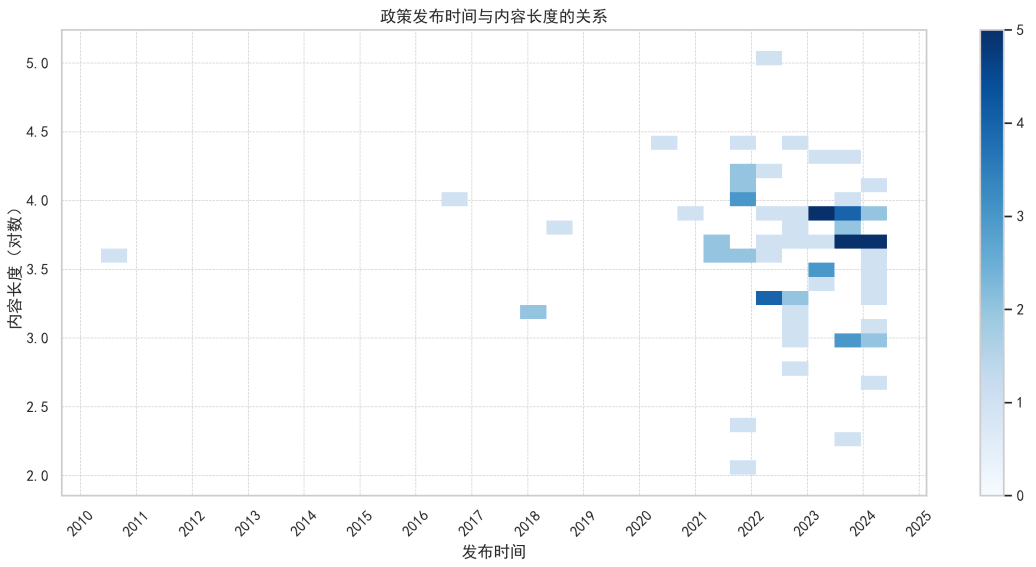

图 3: 政策文本数据集的发布时间与内容长度分布

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 19,335 句，文档平均句子数为 195.3 句，平均字符数 9177.52 字，总字符数达约 90 万汉字，句子平均长度为 64.83 字/句，具备典型政策文风特征（长句、嵌套多、名词短语密集）。数据集描述性信息如表1所示。

表 1: 政策文本数据集描述性信息

指标	值
文档数量	99
总句子数	19335
平均句子数/文档	195.3
平均字符数/文档	9177.52
总字符数	900,000+
平均句子长度（字/句）	64.83

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。

基于 n-gram 的分词策略在中文文本处理中存在一定局限性，这是因为其通过计算每两个字符中间的 PMI（Pointwise Mutual Information）来确定 n-gram 片段的边界，可能导致切分出一些不符合中文语言习惯的片段，尤其是在处理多义词、成语或专有名词时。此外，n-gram 方法在处理长文本时可能会产生大量的冗余片段，增加了模型的计算复杂度和内存消耗。并且，由于没有引入停用词过滤机制，可能会导致一些常见但对语义理解没有实际贡献的词被纳入模型训练中，从而影响模型的性能。相比基于 n-gram 的策略，本文使用的 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。本文使用 HanLP 提供的 api 接口对清洗后的句子进行切词，去除停用词与其他不符合规则（过长或过短）的词，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 $NextSentencePrediction(NSP)$ 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：

$$\mathcal{S} = \{s_1, s_2, \ldots, s_n\} \tag{22}$$

对于每个句子我们利用“逗号”划分子句：

$$s_i = \{s'_{i,1}, s'_{i,2}, \ldots, s'_{i,k}\} \tag{23}$$

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：

$$(s_{i,k}, s_{i,k+1}) \in \mathcal{P} \tag{24}$$

最终得到的正样本对数量为：

$$|\mathcal{P}| = 23374 \tag{25}$$

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：

$$|\mathcal{N}_1| = |\mathcal{P}| = 23374 \tag{26}$$

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）。然而，由于来自不同文章的句子有概率在语义上具有相似性，如两篇介绍地区“数据治理政策”的文章可能都提到“数据治理”这一概念，因此该方案的负样本可能会引入一些噪声，导致模型的训练结果无法很好地反映句子间的真实关系。为此，我们引入了第二个数据集方案。

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，我们设计更复杂的负样本：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：

$$(s_{i,k}, s_{j,l}) \quad \text{其中} \quad j = i + \delta, \quad \delta \in \{2, 3, 4, 5\} \tag{27}$$

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：

$$|\mathcal{N}_2| = 5 \times |\mathcal{P}| = 116870 \tag{28}$$

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。而同一篇文章中相隔较远的句子往往比起随机选取句子更能体现语义上的差异。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 2 所示。

表 2: NSP 任务样本分布

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

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。

3.2.4 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。对 *bge* 模型的处理如图4所示，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。

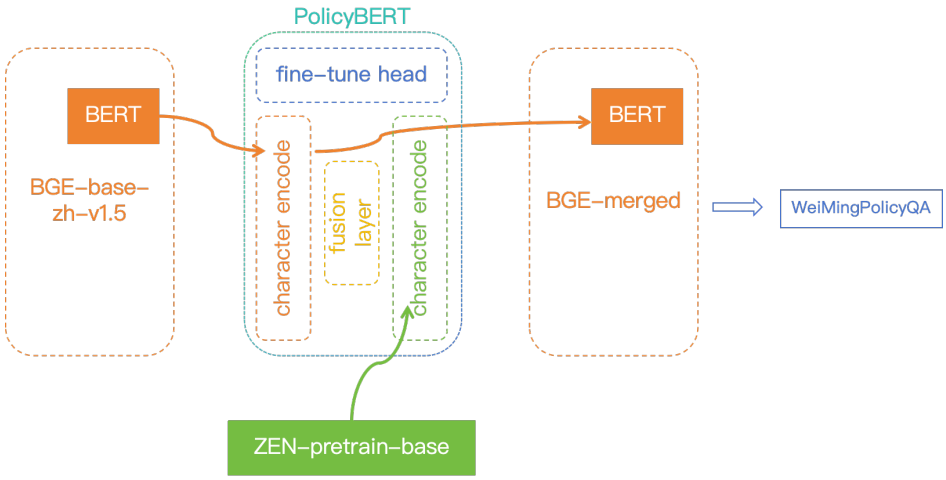

图 4: 对 bge 模型的处理，将其主体 BERT 部分参数提取出来作为骨干字符嵌入模型，其余的参数使用 ZEN 预训练模型进行初始化，微调过的 bge 模型将被用于后续的 RAG 系统中。

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 × 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 公开数据集

首先，我们将在公开数据集上进行实验，与现有的 NLP 领域文本编码器进行对比。包括 BERT-wwm [18], ERNIE 1.0 [8],ERNIE 2.0 [38], NEZHA [39] 和 ZEN [9]。我们使用了 ZEN 的预训练模型作为基线模型进行微调。我们使用了不同的融合方法（门控机制、多头注意力机制）进行对比。由于缺少大规模预训练语料，我们无法在

公开数据集上应用 Hanlp 分词，也无法进行对基线模型的预训练。因此我们在公开数据集上仅测试了不同融合方法与 n-gram 表示的效果。

我们在以下任务上进行了评估：

- **中文分词（CWS）**：使用 SIGHAN2005 中文分词评测（Emerson, 2005）[40] 中的 MSR 数据集。
- **词性标注（POS）**：使用 CTB5 数据集（Xue 等, 2005）[41]，并采用标准数据划分。
- **命名实体识别（NER）**：使用国际中文语言处理评测（Bakeoff 2006）[42] 中的 MSRA 数据集。
- **文档分类（DC）**：使用 THUCNews 数据集（Sun 等, 2016）[43]，该数据集来源于新浪新闻，包含 10 个类别，类别分布均匀。
- **情感分析（SA）**：使用 ChnSentiCorp（CSC）数据集 [44]，该数据集包含来自图书、计算机和酒店三个领域的 12,000 篇文档。
- **句子对匹配（SPM）**：使用 LCQMC 数据集（Liu 等, 2018）[45]，该数据集中的每一个样本都是一个句子对，和一个标签，表示这两个句子是否语义相似。

各个任务的数据集划分和数据集内句子/文档数量如表 3 ：

表 3: 公开数据集上各个任务的数据集划分，包括训练集和测试集的大小

Task	CWS	POS	NER	DC	SA	SPM
Dataset	MSR	CTB5	MSRA	THUCNews	CSC	LCQMC
Train	87K	18K	45K	50K	10K	239K
Test	4K	1K	3K	10K	1K	13K

表 4 展示了不同模型在公开数据集上的 F1 值对比结果，可以看出，PolicyBERT 在多个任务上均表现出色，尤其是在中文分词（CWS）和词性标注（POS）任务中，分别达到了 0.9816 和 0.9716 的 F1 值，显著优于两个基线模型（BERT 和 ZEN）。此外，在句子对匹配（SPM）任务中，PolicyBERT 的多头注意力机制（attn）和门控机制（gate）融合方法分别达到了 0.9016 和 0.8997 的 F1 值，分别为最高和次高的结果，远超过其他所有模型，表明其在捕捉上下文语义关系方面具有明显优势。虽然在

命名实体识别（NER）和文档分类（DC）任务中，PolicyBERT 的表现略逊于一些大模型（如 ERNIE 2.0），但其表现仍然具有竞争力，且在 NER 任务中，PolicyBERT 的注意力机制融合方法（attn）达到了 0.9437 的 F1 值，为次高的结果，显示出其在处理复杂语义关系时的潜力。在文档分类（DC）任务中，PolicyBERT 的表现相对较弱，F1 值为 0.9655，略低于一些大模型（如 ERNIE 2.0 和 NEZHA），但仍然保持在较高水平，显示出其在处理长文本时的稳定性和可靠性。在情感分析（SA）任务中，PolicyBERT 的表现也相对较好，达到了 0.9475 的 F1 值，表明其在处理情感倾向性文本时具有一定的优势。总体而言，PolicyBERT 在多个任务上均展现出良好的性能，尤其是在中文分词和句子对匹配任务中，显示出其在上下文建模和语义理解方面的优势。

表 4: 公开数据集上不同模型的 f1 值对比，其中 B 表示基线模型，L 表示大模型，attn 和 gate 分别表示多头注意力机制和门控机制融合方法。加粗和下划线分别表示在同一列中最好的和次好的结果。

Model\Task	CWS	POS	NER	DC	SA	SPM
BERT	0.9720	0.9543	0.9312	0.9671	0.9410	0.8513
BERT-wwm	-	-	**0.9510**	**0.9760**	0.9500	0.8680
ERNIE 1.0	-	-	0.9510	<u>0.9730</u>	0.9540	0.8740
ERNIE 2.0(B)	-	-	-	-	0.9550	0.8790
NEZHA(B)	-	-	-	-	0.9517	0.8741
NEZHA-wwm(B)	-	-	-	-	<u>0.9584</u>	0.8710
ERNIE 2.0(L)	-	-	-	-	0.9580	0.8790
NEZHA(L)	-	-	-	-	0.9583	0.8720
NEZHA-wwm(L)	-	-	-	-	**0.9600**	0.8794
ZEN	0.9789	0.9582	0.9324	0.9687	0.9442	0.8527
$PolicyBERT_{attn}$	<u>0.9812</u>	<u>0.9711</u>	<u>0.9437</u>	0.9651	0.9441	**0.9016**
$PolicyBERT_{gate}$	**0.9816**	**0.9716**	0.9422	0.9655	0.9475	<u>0.8997</u>

3.3.2 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 5 和表 6 所示。

表 5: PolicySM-1 预训练超参数

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 6: PolicySM-1 下游任务超参数

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

实验结果由表 7 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。

表 7: PolicySM-1 下游任务结果

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**

3.3.3 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 8 和表 9 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。

表 8: PolicySM-2 预训练超参数

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 9: PolicySM-2 下游任务超参数

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

实验结果由表 10 和表 11 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。

表 10: PolicySM-2 预训练结果

骨干模型	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 11: PolicySM-2 下游任务结果

骨干模型	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

4 未名问政

4.1 系统架构

通过在政策文本语料上的微调，本文显著提升了 *bge-base-zh-v*1.5 模型在子领域内对中文语义的理解与表达能力。在此基础上，进一步构建了一个基于检索增强生成（Retrieval-Augmented Generation, RAG）机制的智能问答系统——“未名问政”，旨在为用户提供高效、精准的政策解读与咨询服务，帮助其快速理解并获取所需政策信息。系统整体采用 Langchain-Chatchat 架构，既保证了模块化和可扩展性，又便于集成主流语言模型与检索引擎。系统主要包括检索模块、生成模块和轻量级融合控制层，形成了完整的“查询—检索—生成—响应”流程。

在检索模块中，系统采用基于 PolicyBERT 架构并经过领域微调的 *bge-base-zh-v*1.5 作为中文文本编码器。为进一步融合通用语义能力与子领域知识，本文将微调后的参数与原始模型部分参数进行拼接，构建了 *bge-base-zh-v*1.5-*merge* 模型。具体流程为：首先对政策语料进行分段预处理（每段不超过 750 字），再将其编码为固定维度的向量，存入本地 Faiss 向量数据库。用户输入查询后，系统将查询文本同样编码为向量，并通过相似度检索选出 top-k 个最相关的政策片段，作为生成模块的上下文支持。

在生成模块中，系统通过 Ollama 本地化部署了 $Qwen2.5:7b$ 大语言模型。该模型具备长文本处理能力和优异的中文生成效果，能够结合检索到的政策片段与用户原始问题，通过预设提示模板（Prompt）生成专业、连贯的答案。生成模块将检索模块输出的多个相关片段与用户查询拼接后输入至 $Qwen2.5:7b$，最终输出针对性强、逻辑清晰的回答或政策摘要。

本文还为未名问政系统设计了一个 Logo，如图 5 所示。该 Logo 以“未名问政”

为主题，用“PolicyBERT”体现其背后的微调模型架构。Logo 的设计简洁明了，易于识别，符合现代科技产品的审美趋势。

图 5: 未名问政 Logo

综上，本文构建的“未名问政”系统在本地硬件（RTX 4060 8GB 显存）上完成了从文本检索到问答生成的全流程部署，充分融合了 *bge-base-zh-v*1.5-*merge* 的检索能力与 *Qwen*2.5 : 7*b* 的生成优势，依托 Langchain-Chatchat 架构实现了高度模块化与可配置的中文 RAG 系统。该系统具备部署灵活、响应快速、扩展便捷等特点，尤其适用于政务智能问答、政策知识库构建与自动解读等典型场景。

4.2 系统演示

4.2.1 系统部署

未名问政 RAG 系统的部署过程相对简单，用户只需在本地安装必要的依赖库，并下载预训练模型与知识库数据。系统支持多种操作系统环境，包括 Windows、Linux 和 MacOS。用户可根据需求选择合适的运行环境，并通过命令行界面启动系统。系统的运行效率较高，能够在普通个人电脑上实现快速响应。本次部署配置如下：

- 操作系统：Windows 10
- CPU：Intel Core i9-14900HX
- 内存：32 GB
- 显卡：NVIDIA GeForce RTX 4060 8GB
- Python 版本：3.9.6

部署前应提前配置 Ollama 环境，可于Ollama 官网下载 Ollama，并通过以下代码部署千问模型（Qwen2.5:7b）和微调过的 *bge-base-zh-v*1.5 模型。具体步骤如下：

- 下载模型：在终端中运行以下命令，下载千问模型（Qwen2.5:7b）：

```
ollama pull qwen2.5:7b
```

- 部署微调过的 *bge-base-zh-v*1.5 模型：

```
ollama add bge-base-zh-v1.5 path/to/your/model
```

- 启动 Ollama：在终端中运行以下命令，启动 Ollama 服务并保持运行：

```
ollama serve
```

使用代码将未名问政系统克隆到本地：

```
git clone https://github.com/PuHT4213/WeiMingPolicyRAG.git
cd WeiMingPolicyRAG/lib/chatchat-server
```

创建虚拟环境，并安装必要的依赖库：

```
python3.9 -m venv weimingpolicyrag
source weimingpolicyrag/bin/activate # Linux/MacOS
weimingpolicyrag\Scripts\activate # Windows
pip install -e .
```

确保政策文件在目录 `./libs/chatchat-server/chatchat/data/knowledge_base/WeiMingPolicies/content` 下，然后运行以下命令初始化并启动系统：

```
chatchat kb -r
chatchat start -a
```

出现以下日志以及图6所示的界面后，表示系统已成功启动并运行。用户可在浏览器中访问并进行交互。

```
----------------------------------------------------------
知识库名称          : WeiMingPolicies
知识库类型          : faiss
向量模型:           : bge-large-zh-v1.5
知识库路径          : path/to/your/knowledge_base
文件总数量          : 99
入库文件数          : 99
知识条目数          : 740
用时                : 0:02:29.701002
----------------------------------------------------------
```

```
总计用时          : 0:02:33.414425
```

4.2.2 RAG 对话

如图6所示，未名问政 RAG 系统的主界面简洁直观，当前版本为 1.0.0.1。界面左侧设有功能导航栏，用户可在“多功能对话”、“RAG 对话”与“知识库管理”等主要模块间进行切换。在“RAG 对话”模式下，用户可以根据具体需求，自定义相关配置参数。首先，用户可通过下拉菜单选择对话模式，目前支持“知识库问答”；随后可选择已加载的知识库，以支持特定领域的问答服务。进一步地，系统允许用户设置历史对话轮数（默认为 3 轮）、匹配知识条数（默认为 5 条），以及知识匹配的最低分数阈值（以滑动条形式设置，范围为 0.00 至 2.00，默认值为 0.50），SCORE 越小，相关度越高，取到 2 相当于不筛选。此外，用户还可以勾选“仅返回检索结果”选项，以查看未融合生成的原始检索片段。界面右侧为空白对话窗口区域，用于展示系统响应内容。底部提供输入框，支持多轮对话交互，用户输入内容后可按 Shift+Enter 执行提交，界面整体设计清晰高效，便于快速部署和应用。

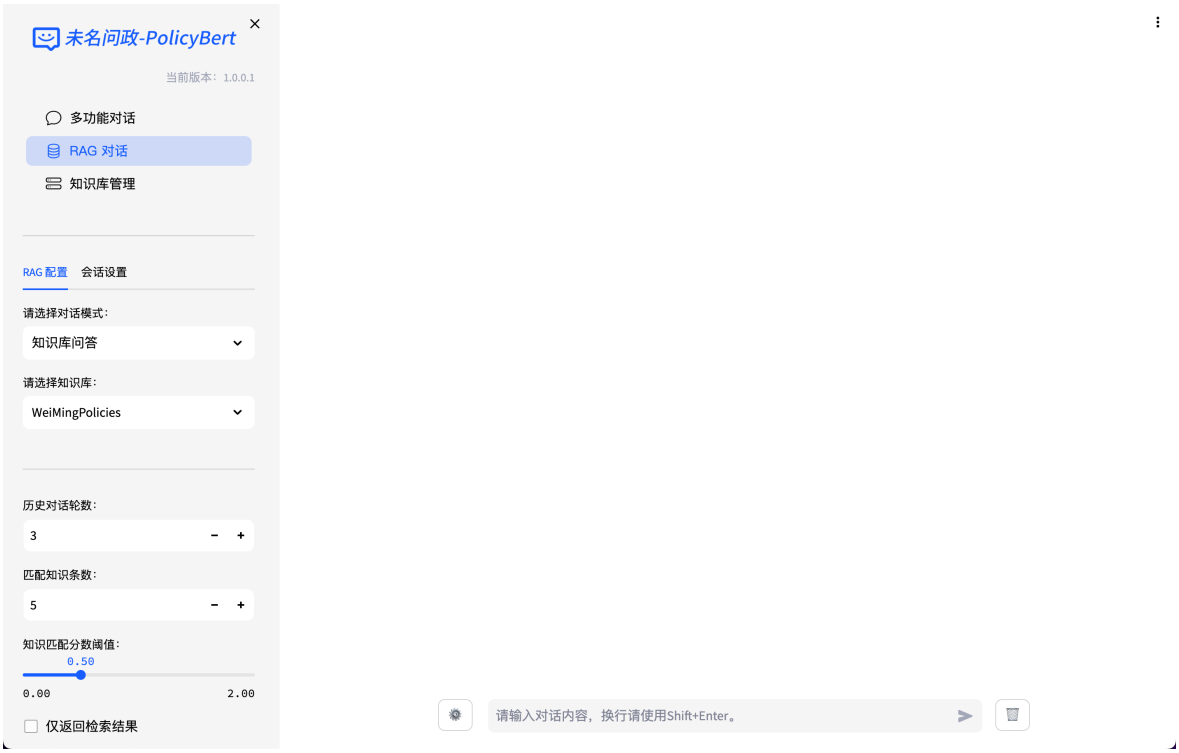

图 6: 未名问政 RAG 系统首页界面。界面左侧为参数配置区域，支持设置对话模式、知识库来源、历史轮数、匹配条数及知识匹配分数阈值，右侧为对话窗口区域，用于显示模型响应结果。该系统通过图形化界面提供了灵活高效的 RAG 任务配置方式。

如图7，未名问政 RAG 在“RAG 对话”模式下提供了图形化的参数配置与实时问答界面。右侧展示了用户提问（如“什么是数据基础制度体系？”）、知识库匹配结果（下拉列表形式展现检索到的原文段落）及模型最终生成的回答。该对话结果展现了未名问政 RAG 系统在政策文本问答任务中的优异表现。系统准确理解了用户提出的“什么是数据基础制度体系？”这一问题，并结合知识库中的相关内容，生成了逻辑清晰、内容详实的回答。回答不仅涵盖了数据基础制度体系的定义，还详细阐述了其包含的四个方面的制度内容，条理分明，层次清晰，充分体现了系统在检索增强生成（RAG）任务中的强大能力。此外，回答语言流畅自然，专业性强，能够很好地满足用户对政策解读的需求，展现了系统在政务智能问答场景中的实际应用价值。

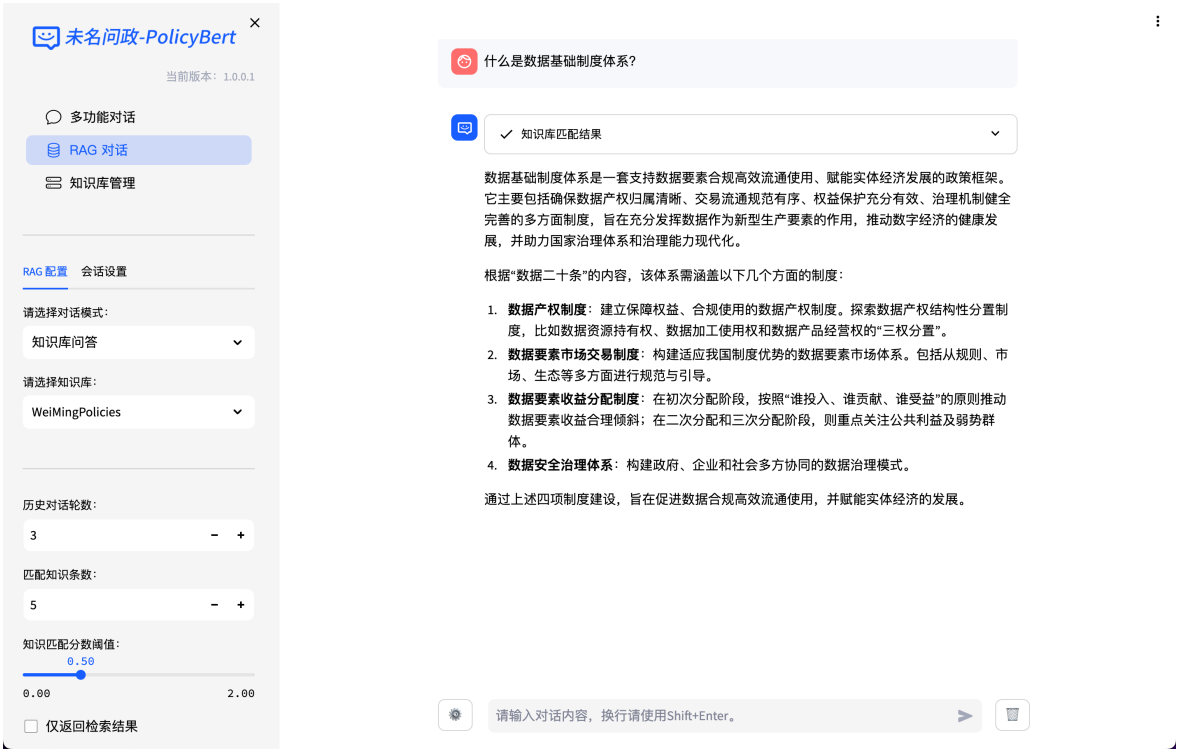

图 7: 未名问政 RAG 系统对话界面。左侧为参数配置区域，右侧为对话窗口区域。用户可在输入框中输入问题，系统将根据设置的参数进行检索与生成。

在图8中的对话中，用户启用了“仅返回检索结果”功能，系统准确地从知识库中检索出了与用户问题“什么是数据基础制度体系？”高度相关的政策文本片段。检索结果内容详实，涵盖了数据基础制度体系的核心定义及其在政策框架中的重要作用，并引用了权威来源，确保了信息的可靠性和专业性。从结果来看，该功能能够有效地将用户问题与知识库中的相关内容进行精准匹配，避免了生成式回答可能带来的信息偏差或冗余问题，特别适用于需要直接引用政策原文或获取权威信息的场景。这种检索增强的方式不仅提升了系统的实用性和可信度，也为用户提供了更高效的政策信息

获取途径。

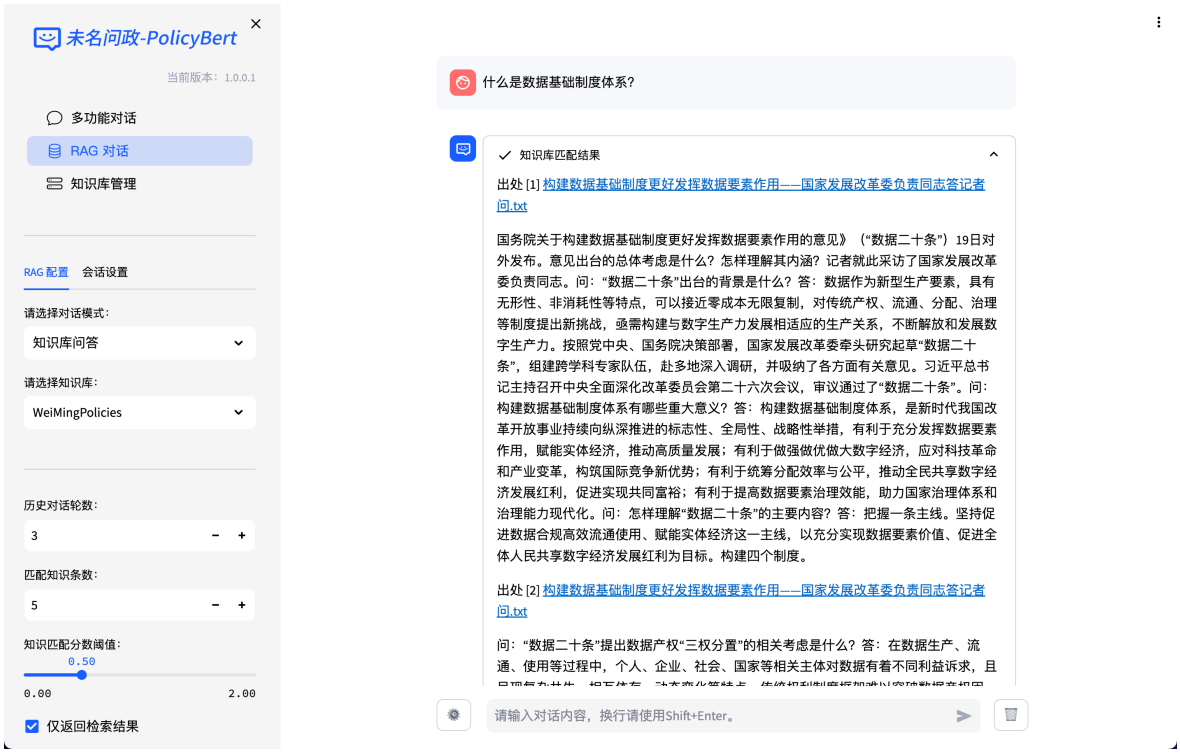

图 8: 未名问政 RAG 系统检索结果界面。用户使用“仅返回检索结果”功能，系统准确地从知识库中检索出了与用户问题“什么是数据基础制度体系？”高度相关的政策文本片段。

4.2.3 知识库管理

如图9和10所示，未名问政 RAG 系统的知识库管理模块允许用户对知识库进行增删改查操作。用户可以通过“上传知识文件”功能，将新的政策文本片段添加至知识库中，以便后续检索与生成使用。同时，系统支持对已有知识进行编辑和删除操作，确保知识库内容的及时更新与维护。此外，用户还可以通过“查看知识”功能，快速浏览当前知识库中的所有条目，方便进行信息检索与管理。该模块的设计旨在提升系统的灵活性与可扩展性，使其能够适应不断变化的政策环境与用户需求。

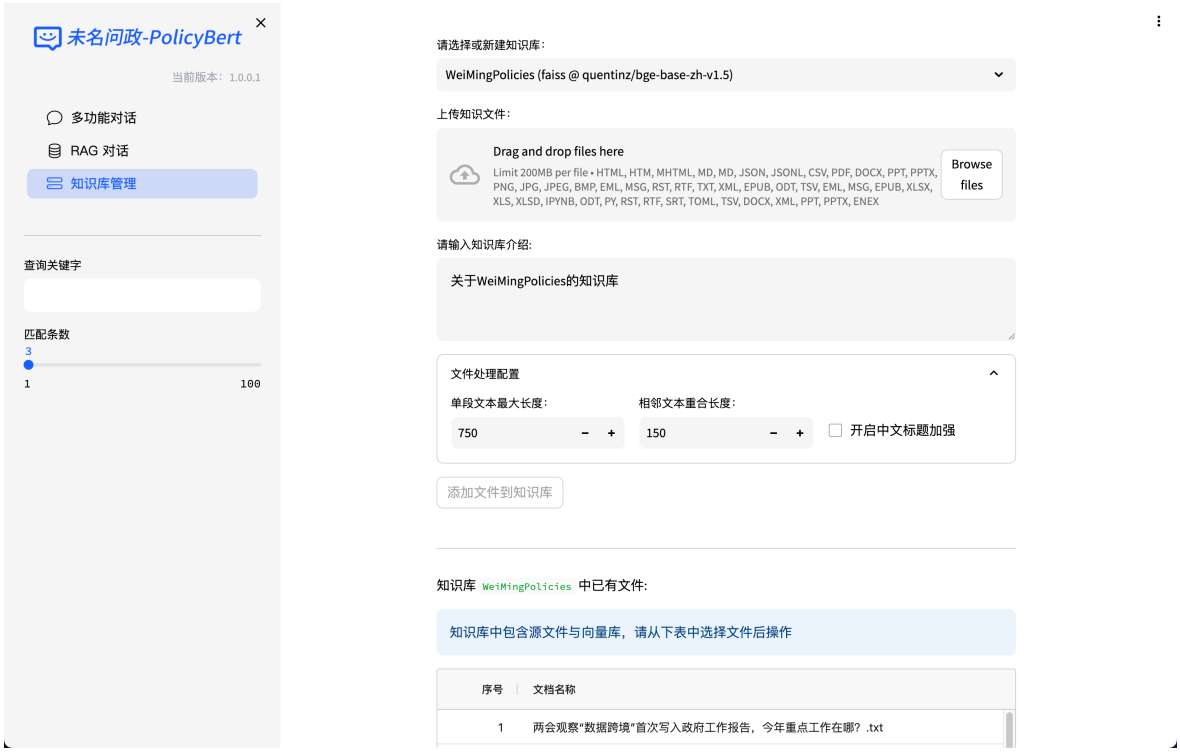

图 9: 未名问政 RAG 系统知识库管理界面。用户可以通过“上传知识文件”功能，将新的政策文本片段添加至知识库中，以便后续检索与生成使用。

图 10: 未名问政 RAG 系统知识库管理界面。用户可以通过“查看知识”功能，快速浏览当前知识库中的所有条目，方便进行信息检索与管理。

5 总结与展望

5.1 结论与总结

本文提出了一种基于融合式中文编码模型的文本编码器，旨在提升中文政策文本的语义理解与生成能力。通过对比不同的融合方法（门控机制、多头注意力机制）与分词策略（n-gram、Hanlp），我们验证了融合式模型在中文政策文本处理中的有效性。实验结果表明，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。此外，我们还构建了两个适用于中文政策文本的 NSP 任务数据集（PolicySM-1 和 PolicySM-2），并在这两个数据集上进行了模型微调。实验结果显示，使用多头注意力机制的模型在 NSP 任务中表现最佳，达到了 0.9325 的准确率。最后，基于微调后的模型，我们构建了一个基于检索增强生成（RAG）机制的智能问答系统“未名问政”，实现了从文本检索到问答生成的全流程部署。该系统具备部署灵活、响应快速、扩展便捷等特点，适用于政务智能问答、政策知识库构建与自动解读等场景。

5.2 不足与展望

尽管本文提出的 PolicyBERT 模型在中文政策文本的语义理解与生成任务中取得了显著的性能提升，但仍存在一些不足之处需要进一步改进。首先，模型的训练和评估主要基于特定领域的政策文本数据集，虽然在该领域表现优异，但其在其他领域的泛化能力尚未得到充分验证。未来可以尝试扩展数据集的覆盖范围，探索模型在多领域文本处理中的适用性。

本文采用了 HanLP 分词工具和基于门控机制与多头注意力机制的融合方法，尽管在实验中表现出色，但分词错误和词表构建的局限性可能对模型性能产生一定影响。未来可以尝试引入更先进的分词技术或预训练方法，以进一步提升模型对中文语言特性的适应能力。

此外，模型的训练和评估主要集中在 NSP 任务上，虽然在该任务上取得了良好的效果，但在其他下游任务（如文本分类、情感分析等）上的表现尚未得到充分验证。未来可以尝试构建更多的下游任务数据集，并对模型进行多任务学习，以提升其在不同任务上的泛化能力。

由于缺少大规模预训练语料，模型无法进行在大规模语料上从零开始的预训练，导致模型在一些复杂的语义理解任务上仍然存在一定的局限性。未来可以尝试引入更多的预训练数据，或结合其他预训练模型进行迁移学习，以提升模型的性能。

最后，模型的计算复杂度较高，尤其是在多头注意力机制的融合过程中，对硬件

资源的需求较大。这在一定程度上限制了模型在资源受限环境中的应用。未来可以尝试优化模型结构或引入轻量化技术，如知识蒸馏或模型剪枝，以降低计算成本并提升部署效率。

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.

[11] Han He and Jinho D. Choi. The stem cell hypothesis: Dilemma behind multi-task learning with transformer encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5555–5577, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[12] Robert Lakatos, Peter Pollner, Andras Hajdu, and Tamas Joo. Investigating the performance of retrieval-augmented generation and fine-tuning for the development of ai-driven knowledge-based systems, 2024.

[13] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

[14] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang

Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[15] Qian Liu, Jinke Song, Zhiguo Huang, Yuxuan Zhang, glide the, and liunux4odoo. langchain-chatchat. `https://github.com/chatchat-space/Langchain-Chatchat`, 2024.

[16] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2017.

[17] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.

[18] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514, 2021.

[19] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. Revisiting pre-trained models for chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 2020.

[20] Dongling Xiao, Yu-Kun Li, Han Zhang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-gram: Pre-training with explicitly n-gram masked language modeling for natural language understanding, 2021.

[21] Xinnian Liang, Zefan Zhou, Hui Huang, Shuangzhi Wu, Tong Xiao, Muyun Yang, Zhoujun Li, and Chao Bian. Character, word, or both? revisiting the segmentation granularity for chinese pre-trained language models, 2023.

[22] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language rep-

resentation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, September 2019.

[23] Taolin Zhang, Zerui Cai, Chengyu Wang, Minghui Qiu, Bite Yang, and Xiaofeng He. Smedbert: A knowledge-enhanced pre-trained language model with structured semantics for medical text mining, 2021.

[24] Guangyi Liu, Yinghong Liao, Fuyu Wang, Bin Zhang, Lu Zhang, Xiaodan Liang, Xiang Wan, Shaolin Li, Zhen Li, Shuixing Zhang, and Shuguang Cui. Medical-vlbert: Medical visual language bert for covid-19 ct report generation with alternate learning. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9):3786–3797, September 2021.

[25] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Legal-bert: The muppets straight out of law school, 2020.

[26] Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. Lawformer: A pre-trained language model for chinese legal long documents, 2021.

[27] Haoxi Zhong, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. Open chinese language pre-trained model zoo. Technical report, THUNLP, 2019.

[28] Bihui Yu, Chen Deng, and Liping Bu. Policy text classification algorithm based on bert. In *2022 11th International Conference of Information and Communication Technology (ICTech))*, pages 488–491, 2022.

[29] Kaixin Ma, Meiling Liu, Tiejun Zhao, Jiyun Zhou, and Yang Yu. Ptwa: Pre-training with word attention for chinese named entity recognition. *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021.

[30] Wenbiao Li, Rui Sun, and Yunfang Wu. Exploiting word semantics to enrich character representations of chinese pre-trained models. In *Natural Language Processing and Chinese Computing*, 2022.

[31] Qiang He, Guowei Chen, Wenchao Song, and Pengzhou Zhang. Prompt-based word-level information injection bert for chinese named entity recognition. *Applied Sciences*, 2023.

[32] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing,

Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[33] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

[34] Xuejiao Zhao, Siyan Liu, Su-Yin Yang, and Chunyan Miao. Medrag: Enhancing retrieval-augmented generation with knowledge graph-elicited reasoning for healthcare copilot, 2025.

[35] Muhammad Rafsan Kabir, Rafeed Mohammad Sultan, Fuad Rahman, Mohammad Ruhul Amin, Sifat Momen, Nabeel Mohammed, and Shafin Rahman. Legal-rag: A hybrid rag system for multilingual legal information retrieval, 2025.

[36] Xinyu Wang, Jijun Chi, Zhenghan Tai, Tung Sum Thomas Kwok, Muzhi Li, Zhuhong Li, Hailin He, Yuchen Hua, Peng Lu, Suyuchen Wang, Yihong Wu, Jerry Huang, Jingrui Tian, and Ling Zhou. Finsage: A multi-aspect rag system for financial filings question answering, 2025.

[37] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024.

[38] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 2.0: A continual pre-training framework for language understanding, 2019.

[39] Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. Nezha: Neural contextualized representation for chinese language understanding, 2021.

[40] Thomas Emerson. The second international Chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, 2005.

[41] Naiwen Xue, Fei Xia, Fu-dong Chiou, and Marta Palmer. The penn chinese treebank: Phrase structure annotation of a large corpus. *Nat. Lang. Eng.*, 11(2):207–238, June 2005.

[42] Gina-Anne Levow. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, Sydney, Australia, July 2006. Association for Computational Linguistics.

[43] M. Sun, J. Li, Z. Guo, Z. Yu, Y. Zheng, X. Si, and Z. Liu. Thuctc: An efficient chinese text classifier. `https://github.com/thunlp/THUCTC`, 2016. GitHub Repository.

[44] Songbo Tan. Chnsenticorp, 2020.

[45] Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. LCQMC:a large-scale Chinese question matching corpus. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

6 致谢

感谢孟凡老师一直以来的指导与支持。从大二怀着忐忑初次走进组会时起，您总能用平实的话语拨开我心头的雾霭。从论文开题到实验设计，每一次指导与反馈，都是一盏为我照亮前路的明灯。因为您的支持，我才得以从稚嫩中走出，以独立之姿完成毕业设计，写出这篇论文。

感谢组内所有同学。我们在一次次组会讨论中磨砺思维，在协作中互补长短。正是这些或激烈、或温柔的碰撞，让我看见自己的盲区，也体味到团队协作的真谛。几年的相知相伴，使我深刻懂得：学术之路，不只是个人的修行。

感谢晋康玮同学，你不仅是青葱岁月里的知己，更是我一路走来的同行者。从高中到大学，无数次在我迷茫与失落之际，是你的陪伴给我力量。

更要感谢我的父母，你们用无声的支持与深沉的爱，为我遮风挡雨。无论身在何处，你们永远是我心灵的港湾，是我追逐远方时最温暖的牵挂。

感谢因元火而相识的同学们，我们因共同的热爱而相聚，虽然关系有近有远，但我们度过的无数日夜，一次次彻夜畅谈，一次次踏星而归，都是我心中最珍贵的回忆，与前行的动力。

回望这四年，每一步都刻下成长的印记。大一时的懵懂，大二时的挣扎，大三时的蜕变，大四时的解脱。四年很长，但又好像一瞬间。或许前路依旧未知，但正是这些经历，铸就了如今的我。

因此，最后，我要感谢我自己，从不后悔做过的每一个选择，在无数挫折中一次又一次地站起，歌唱勇气。

7 北京大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名：_____

日期：_____ 年 _____ 月 _____ 日

学位论文使用授权说明

本人完全了解北京大学关于收集、保存、使用学位论文的规定，即：

- 按照学校要求提交学位论文的印刷本和电子版本；
- 学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务，在校园网上提供服务；
- 学校可以采用影印、缩印、数字化或其它复制手段保存论文。

论文作者签名：_____ 导师签名：_____

日期：_____ 年 _____ 月 _____ 日