****



**研 究 生 毕 业 论 文**

**（申请工程硕士学位）**

|  |  |
| --- | --- |
| **论文题目** | 基于深度学习的法条推荐技术研究 |
| **作者姓名** | 曾 进 |
| **学科、专业名称** | 工程硕士(软件工程领域) |
| **研究方向** | 软件工程 |
| **指导教师** | 葛季栋　副教授 |

**2019年 6 月 20 日**

**学 号： MF1732007**

**论文答辩日期： 年 月 日**

**指 导 教 师： （签字）**

**基于深度学习的法条推荐技术研究**

|  |  |
| --- | --- |
| **作 者:** | **曾 进** |
| 指导教师: | **葛季栋　副教授** |

|  |
| --- |
| **南京大学研究生毕业论文** |
| **(申请工程硕士学位)** |

|  |
| --- |
| **南京大学软件学院** |
| **2019年6月** |

**Research on Statutes**

**Recommendation Based on**

**Deep Learning**

**Zeng, Jin**

**Submitted in partial fulfillment of the requirements for the degree of Master of Engineering**

Supervised by

Associate Professor **Ge, Jidong**

Software Institute

**NANJING UNIVERSITY**

Nanjing, China

June, 2019

# 摘 要

在当下社会矛盾日益增多的情况下，大量司法案件涌入法院，而司法资源本身的局限性，导致法官面临严重的工作压力。在全面推进人民法院信息化建设背景下，大量法律裁判文书信息公开化，“法律人工智能”成为研究焦点。通过对裁判文书的挖掘与分析，不仅能为法院提供更加智能的技术来提高办公效率，还能为公众提供智能的法律咨询服务，促进司法可得。法条推荐作为实现这些目标的技术之一，一方面能为法官推荐在审判时可能适用的法条，提高工作效率；另一方面还能帮助当事人跨越司法鸿沟，提前了解案件审判倾向和相关法条，并形成最佳诉讼策略。

本文详细分析了法条推荐问题，并介绍了法律裁判文书的数据特点，包括种类繁多、有较多法律领域专用词以及半口语化等，仅仅从字符匹配层面难以实现真正智能化的服务。所以本文根据案件基本情况与引用法条关联紧密的特点，重点对裁判文书案件基本情况进行语义分析，以深度学习方法为基础，以XML格式的裁判文书为数据源，展开了研究工作。

首先，本文提出了针对裁判文书特点的预处理方法，包括法条标准化、案由标准化、分词、建立法律专用停用词以及去停用词。接着提出两种法条推荐方法，分别是基于LSTM(Long short-term memory)的法条推荐方法以及基于LDA-LSTM(Latent Dirichlet Allocation, Long short-term memory)的法条推荐方法，具体如下：

（一） 基于LSTM的法条推荐方法：法官在判案时经常会参考案件基本情况的内容，其内容与引用的法条实际上有很大关联。该方法通过对输入的案件基本情况序列进行语义分析，挖掘案件基本情况与法条的内在联系。本文通过动态LSTM对变长的输入序列进行处理，利用深度学习方法获得案件基本情况的语义向量，并基于该语义向量进行多标签多分类，完成法条推荐任务。

（二）基于LDA-LSTM的法条推荐方法：通过模拟人类注意力，即法官在阅读文书时会额外关注与文书主题相关的词语，来对案件进行审判。该方法通过引入案件基本情况的主题向量作为先验信息，利用Attention机制(Attention Mechanism)，在计算注意力向量时赋予与主题相关的词更大的权重，并基于计算出的注意力向量进行多标签多分类，完成法条推荐任务。

在实验验证阶段，本文以六种民事案由的裁判文书为数据集，设计了一系列对比实验。其中对比了基于深度学习的法条推荐方法与基于传统机器学习LightGBM的法条推荐方法的推荐效果，验证了深度学习自主学习特征以及加入LDA先验信息的有效性。最后对比了所有方法的优缺点，并对未来的研究工作进一步展望。

**关键词：**法条推荐、裁判文书、深度学习、多标签多分类、长短期记忆网络、注意力机制、词向量模型、主题模型

# Abstract

In the current situation of increasing social disputes, a large number of judicial cases have flooded into the courts, and the limitations of judicial resources themselves have caused judges to face severe work pressure. Under the background of comprehensively promoting the informatization construction of the people's courts, a large number of legal judgment documents have been publicized, and "legal artificial intelligence" has become the focus of research. Through the excavation and analysis of the judgment documents, the court can provide more intelligent technology to improve office efficiency, and it can help to provide intelligent legal counseling services to the public and promote judicial access. The Statutes Recommendation is regarded as one of the techniques to achieve these goals. On the hand, it can recommend usful statutes for the judges, and improve the efficiency of work; on the other hand, it can help the parties to cross the judicial gap and understand the trial tendency and related statutes in advance, and form the best litigation strategy.

This thesis analyzes the Statutes Recommendation in detail, and introduces the data characteristics of the legal judgment documents, including a wide variety, more legal domain specific words and semi-verbalization, so it is difficult to realize truly intelligent services only from the character matching level. Based on the close relationship between the basic situation and the reference statutes, this thesis focuses on the semantic analysis of the basic situation of the judgment document case. Based on the deep learning method, the research work is carried out with the XML format judgment document as the data source.

First of all, this thesis proposes a pre-processing method for the characteristics of the referee documents, including standardization of the statutes, standardization of the case, word segmentation, establishment of legal-specific stop words, and de-stopping. Then, two methods of the Statutes Recommendation are proposed, which are based on the LSTM (Long short-term memory) method and the LDA-LSTM (Latent Dirichlet Allocation, Long short-term memory) method.

(1) Statutes recommendation method based on LSTM: The judge often refers to the content of the basic situation of the case when judging the case, and its content is actually related to the cited statutes. The method analyzes the basic situation of the case and the internal relationship between the case and the statutes. In this thesis, the variable length input sequence is processed by dynamic LSTM, and the semantic vector of the basic case is obtained by using the deep learning method. Based on the semantic vector, multi-label and multi-classification are performed to complete the task of statutes recommendation.

(2) Statutes recommendation method based on LDA-LSTM: By simulating human attention, the judge will additionally pay attention to the words related to the subject of the instrument when reading the document to judge the case. By introducing the subject vector of the basic case of the case as a priori information, the Attention Mechanism is used to assign greater weight to the topic-related words when calculating the attention vector, and based on the calculated attention vector, multi-label and multi-classification are performed to complete the task of statutes recommendation.

In the experimental verification stage, this thesis designs a series of comparative experiments with the judging documents of six civil cases as the data set. The recommendation effect of the deep learning-based Statutes Recommendation method and the traditional machine learning LightGBM-based Statutes Recommendation method based on LSTM compared, and the effectiveness of deep learning autonomous learning and the validity of adding LDA prior information is verified. Finally, the advantages and disadvantages of all methods are compared, and the future research work is further prospected.

**Keywords:** Associated Statutes Recommendation, Judgment Documents, Deep Learning, Multi-Label Multi-Classification, Long Short-Term Memory Network, Attention Mechanism, Word2Vec, Topic Model

# 目 录

[摘 要 I](#_Toc3559722)

[Abstract III](#_Toc3559723)

[目 录 V](#_Toc3559724)

[图目录 VII](#_Toc3559725)

[表目录 VIII](#_Toc3559726)

[第一章 引言 1](#_Toc3559727)

[1.1 选题的背景与意义 1](#_Toc3559728)

[1.2 国内外研究现状 3](#_Toc3559729)

[1.3 本文研究内容与贡献 5](#_Toc3559730)

[1.4 本文的组织结构 6](#_Toc3559731)

[第二章 相关背景知识 7](#_Toc3559732)

[2.1 多标签多分类学习 7](#_Toc3559733)

[2.2 非均衡样本学习 7](#_Toc3559734)

[2.3 Word2Vec模型 8](#_Toc3559735)

[2.4 LDA主题模型算法 12](#_Toc3559736)

[2.5 LSTM深度网络与Attention机制 14](#_Toc3559737)

[2.5.1 LSTM深度网络 15](#_Toc3559738)

[2.5.2 Attention机制 18](#_Toc3559739)

[2.6 本章小结 22](#_Toc3559740)

[第三章 法条推荐方法设计 23](#_Toc3559741)

[3.1 法条推荐系统总体框架 23](#_Toc3559742)

[3.1.1 法条推荐问题分析 24](#_Toc3559743)

[3.1.2 裁判文书数据特点 24](#_Toc3559744)

[3.2 法条推荐系统总流程 25](#_Toc3559745)

[3.3 数据预处理 27](#_Toc3559746)

[3.3.1 裁判文书读取 27](#_Toc3559747)

[3.3.2 案由和法条标准化 28](#_Toc3559748)

[3.3.3 分词、去停用词及法律专用停用词 29](#_Toc3559749)

[3.4 基于LSTM的法条推荐方法 31](#_Toc3559750)

[3.4.1 具体流程图 32](#_Toc3559751)

[3.4.2 模型结构及训练 33](#_Toc3559752)

[3.4.3 代码实现 36](#_Toc3559753)

[3.5 基于LDA-LSTM的法条推荐方法 37](#_Toc3559754)

[3.5.1 具体流程图 38](#_Toc3559755)

[3.5.2 模型结构及训练 39](#_Toc3559756)

[3.5.3 代码实现 41](#_Toc3559757)

[3.6 本章小结 41](#_Toc3559758)

[第四章 实验分析 43](#_Toc3559759)

[4.1 数据集 43](#_Toc3559760)

[4.2 评估标准 44](#_Toc3559761)

[4.3 预训练LDA与Word2vec模型 47](#_Toc3559762)

[4.3.1 预训练Word2vec词向量模型 47](#_Toc3559763)

[4.3.2 预训练LDA主题模型 48](#_Toc3559764)

[4.4 法条推荐模型的超参数设置 49](#_Toc3559765)

[4.4.1 基于LSTM的法条推荐方法超参数设置 49](#_Toc3559766)

[4.4.2 基于LDA-LSTM的法条推荐方法超参数设置 52](#_Toc3559767)

[4.5 对比实验 53](#_Toc3559768)

[4.5.1 标准化标签集中法条总数对推荐效果的影响 54](#_Toc3559769)

[4.5.2 法条推荐个数K对推荐效果的影响 55](#_Toc3559770)

[4.5.3 预训练词向量对LSTM、LDA-LSTM的影响 57](#_Toc3559771)

[4.5.4 LSTM、LDA-LSTM与LightGBM方法实验对比 57](#_Toc3559772)

[4.5.5 多分类、多标签多分类方法的对比 59](#_Toc3559773)

[4.6 方法的优缺点分析 60](#_Toc3559774)

[4.7 本章小结 61](#_Toc3559775)

[第五章 总结和展望 62](#_Toc3559776)

[5.1 总结 62](#_Toc3559777)

[5.2 展望 62](#_Toc3559778)

[参 考 文 献 64](#_Toc3559779)

[致 谢 67](#_Toc3559780)

[附 录 68](#_Toc3559781)

[参 与 项 目 69](#_Toc3559782)

[版权及论文原创性说明 70](#_Toc3559783)

# 图目录

[图2. 1 CBOW模型 10](#_Toc3065486)

[图2. 2 基于Hierarchical Softmax的CBOW模型 11](#_Toc3065487)

[图2. 3 LDA概率图模型 13](#_Toc3065488)

[图2. 4 LSTM展开后网络 16](#_Toc3065489)

[图2. 5 LSTMCell内部结构图 16](#_Toc3065490)

[图2. 6 LSTM文本生成 18](#_Toc3065491)

[图2. 7 基于LSTM的seq2seq模型 19](#_Toc3065492)

[图2. 8 基于LSTM与Attention机制的seq2seq模型 21](#_Toc3065493)

[图3. 1 法条推荐系统内部框架图 23](#_Toc3067579)

[图3. 2 法条推荐系统内部流程图 26](#_Toc3067580)

[图3. 3 裁判文书信息提取结果示意图 28](#_Toc3067581)

[图3. 4 基于LSTM的法条推荐方法流程图 32](#_Toc3067582)

[图3. 5 基于LSTM的法条推荐方法模型结构图 35](#_Toc3067583)

[图3. 6 基于LSTM的法条推荐方法模型构建代码 37](#_Toc3067584)

[图3. 7 基于LSTM的法条推荐方法模型训练代码 37](#_Toc3067585)

[图3. 8 基于LDA-LSTM的法条推荐方法流程图 38](#_Toc3067586)

[图3. 9 基于LDA-LSTM的法条推荐方法模型结构图 39](#_Toc3067587)

[图3. 10 Attention代码实现 40](#_Toc3067588)

[图4. 1 Word2vec模型的训练 46](#_Toc3065563)

[图4. 2 Word2vec模型测试结果 46](#_Toc3065564)

[图4. 3 LDA主题模型的训练 47](#_Toc3065565)

[图4. 4 LDA主题下词分布示意图 47](#_Toc3065566)

[图4. 5 参数pos\_weight对模型的影响 50](#_Toc3065567)

[图4. 6 计算Attention的空间对模型的影响 52](#_Toc3065568)

[图4. 7标签总数N对法条推荐效果的影响 53](#_Toc3065569)

[图4. 8 K值对模型CoverLaw的影响 54](#_Toc3065570)

[图4. 9 K值对模型EvalLaw的影响 55](#_Toc3065571)

[图4. 10 多标签多分类下不同模型运行结果对比图 57](#_Toc3065572)

[图4. 11多分类下不同模型运行结果对比图 57](#_Toc3065573)

# 表目录

[表3. 1 案件以及引用法条 24](#_Toc3065769)

[表3. 2 各节点path路径 27](#_Toc3065770)

[表3. 3 法条标准化前后对照表 29](#_Toc3065771)

[表3. 4 法条引用频次统计 29](#_Toc3065772)

[表3. 5 分词前后对照表 31](#_Toc3065773)

[表3. 6 法条label经one-hot处理前后对照表 35](#_Toc3065774)

[表4. 1 裁判文书数据集统计表 42](#_Toc3067601)

[表4. 2 切分训练集与测试集对照表 43](#_Toc3067602)

[表4. 3 混淆矩阵 43](#_Toc3067603)

[表4. 4 多分类问题分解为二分类对照表 44](#_Toc3067604)

[表4. 5 LDA参数设置 47](#_Toc3067605)

[表4. 6 基于LSTM的法条推荐方法超参数设置 48](#_Toc3067606)

[表4. 7 max\_time\_step\_size与显存占用关系表 49](#_Toc3067607)

[表4. 8 基于LDA-LSTM的法条推荐方法超参数设置 50](#_Toc3067608)

[表4. 9 预训练词向量对CoverLaw的影响 55](#_Toc3067609)

[表4. 10 预训练词向量对num\_batch的影响 56](#_Toc3067610)

[表4. 11 多分类与多标签多分类方法对比 58](#_Toc3067611)

# 第一章 引言

## 1.1 选题的背景与意义

近几年，最高人民法院围绕着全面依法治国的战略部署，以“大数据、大格局、大服务”为理念，大力推进人民法院信息化建设。而裁判文书作为记录法院审理过程以及结果的主要工具，已经被公认为是诉讼活动结果的载体，也是确定和分配当事人权力义务的唯一凭证。据统计，截止到目前为止，中国裁判文书网公开的裁判文书已经超过4600万篇，这为法律人工智能研究提供了强有力的数据支持。

在当下数据驱动的大数据时代，由于得到充足的数据，许多司法大数据的研究也相继展开，“人工智能”技术更是广泛应用到裁判文书的分析中。通过挖掘裁判文书中有价值的信息，采用智能化的手段，可以让法律相关的行业运作事半功倍。对于律师而言，繁杂的法律业务诸如法条检索、法律写作可以通过具有智能交互界面的系统完成；对于法院而言，司法审判的数字化和信息化，将有效提高法院办公效率以及确保司法的公信公正；对于当事人而言，可以在没有任何法律认知的情况下，大致了解涉事案件的审判结果。举例如下：

（1）通过人工智能技术推动法律文件的自动化：利用NLP以及机器学习的技术完成案件电子材料的收集与整理，大大节约了审阅文书的时间，也提高了法院的工作效率，同时有助于促进智能机器辅助甚至独立起草法律文件。

（2）基于人工智能技术有助于帮助当事人选择最佳的诉讼策略：基于案件的基本情况，向当事人或律师推荐过去类似的案件或者适用的法律法条作为参考，拟定最佳的诉讼策略。另一方面，当事人不需要咨询律师就可以在案件起诉前对案件胜诉的可能性进行评估，如果可以较为准确的预知案件审判结果，那么当事人也不用冒着可能败诉的风险继续上诉，从某种角度来说，缓解了法院的工作压力，也节约了当事人诉讼成本。

（3）基于人工智能技术有助于快速解决庭审纠纷：通过案件的基本陈述，构建模型来预测案件的审判结果，为法院节省了劳动力，也为法官减轻了工作压力。

其中，通过法律文书的文本挖掘技术，提供更加智能的技术手段辅助法官完成案件的审判，或者帮助没有法律经验的人群更加便捷地了解某案件应该会如何审判。在案件审判过程中为法官推荐可能适用的法律法条，提高法官的办案效率以及维护司法审判的公平公正，则是本文的研究重点。完成法条推荐是非常有意义的，列举以下三点：

（1）通常，法院在进行案件审判的过程中，法官需要结合当事人的诉求、查明的事实以及证据，阅读大量的法律条文才能完成对案件的审判。由于成文法繁多，同样的庭审矛盾可能会出现在不同的法律规范中，法官需要阅读大量的法律法规以及类案的审判，才能给出最合适的判决结果，但这是一件费时费力的事情。并且在当下社会矛盾日益增多的情况下，法院有限的资源难以承受如此大的工作压力。

（2）由于在庭审时法官的主观因素，通常会出现“同案不同判，法律适用结果不统一”的结果，这会影响到司法的公平公正，导致当事人缠诉，影响法院的办公效率。

（3）对于案件的当事人，往往想要第一时间了解案件会如何审判，但因为缺乏法律常识与经验，不得不支付高昂的律师咨询费以帮助了解案件的细节。

法条推荐选题的价值正体现在上述三个方面：为法官推荐可能适用法律法条，提高法官的工作效率；避免因为复杂的社会关系导致的案件误判、偏袒，维护司法的公平公正；帮助当事人了解同类案件的审判结果以及适用法条，形成最佳的诉讼策略，节约咨询律师与诉讼的成本。

继2016年Google的围棋机器人AlphaGo之后，法律人工智能就开始掀起了一番浪潮。近几年，端到端的深度学习方法在法律文本分类、文本摘要以及自然语言案例检索都有了很广泛的应用。由于有了强有力的数据支持，可以有效的将深度学习方法应用到法律文本分析中。利用端到端的深度学习方法，加入已知的LDA先验信息，通过有监督的方式，能够挖掘裁判文书审判的内在规律和裁判文书的语义信息。由于诉讼种类多样，裁判文书半结构化以及半口语化的特点，不仅需要从词汇层级上对法律文书进行分析，还需要从语义层级上对法律文书进行挖掘。因此本文以端到端的深度学习方法为基础，提出了两种基于深度学习的法条推荐方法，并以裁判文书为数据源，着重研究了基于LDA-LSTM的法条推荐方法。

## 1.2 国内外研究现状

当前我国司法大数据应用广泛，结合云计算、大数据分析与人工智能技术辅助裁判等科技创新层出不穷，成为提升治理社会、治理犯罪能力的推动力量。例如：在司法公开背景下大力推进裁判文书上网工作；利用大数据技术进行犯罪信息与趋势预测，避免出现更多的社会矛盾；运用大数据技术建立案件类别占比，确定法官的工作量大小，进行科学的任务分配；基于大数据条件展开的人工智能技术应用，比如类案推荐、量刑辅助以及法条推荐等。

我国对于司法大数据与“智慧法院”的瓶颈正处于数据分析阶段，与其他领域的数据结构有所不同，法律裁判文书主要是结构化与半结构化的数据，这些数据通常是半规范化的自然语言描述。但是难点在于，即使能够发现法律文书中固有的词汇层级描述模式，但是这样的模式无法通过人工的方式完整正确提取出来，更多的需要去关注裁判文书中隐含的语义信息。目前通常的研究方法是通过人工构建数学模型，利用机器学习算法挖掘隐藏在裁判文书中的语义信息来完成法律文书的信息挖掘。在人工智能领域，目前作为主流的“监督学习”方式，采用何种算法、使用数据的质量高低将直接决定最终效果。算法的核心主要在于从数据中正确认识、提炼和总结一般规律，以此归纳出正确的模型，并用于预测未来可能的审判结果。

从事法学研究工作以及从事大数据分析和人工智能技术研究的人都相继提出了不同的方法，对推动司法智能做出了重大贡献。郭叶等[郭叶等，2017]将部分指导性案例作为研究对象，研究数据来源于“北大法宝-司法案例库”中的大量裁判文书，深入剖析了司法应用的特点与规律，对司法应用中可能存在的问题给出了针对性的意见。王竹[王竹，2017]经研究发现，可以利用大数据分析技术辅助分析民法总则的法律规范的去留问题，总结出一系列适用于民法一般规律。张保生[张保生，2001]从法律推理与人工智能角度研究了人工智能法律体系对法学理论与法律实践的价值与意义，其中着重研究了辅助司法审判的重要影响。康东[康东，2014]以中文文本为研究对象，以LTP系统为分词工具，利用词向量工具提取出特征后，再通过支持向量机算法完成中文文本多分类的任务。简璐瑶[简璐瑶，2012]借助Stanford大学本体构建“七步法”，构建了基于领域本体的法律文本挖掘模型，实现了从非结构化的裁判文书提取信息。

从本质上而言，对于法律文书的挖掘还是隶属于文本挖掘领域，因此有必要对文本挖掘的研究进展进行简单陈述。王继成等[王继成等，2000]研究了Web文本挖掘的方法，包括文本的特征向量表示、文本分类以及文本聚类，提出了更快速更准确挖掘Web的HTML文档的方法。李凡等[李凡等，2001]就提取文本特征向量的TFIDF算法加以改进，提出一种可以代替IDF函数的新算法。王海亮[王海亮，2017]结构化存储海量法律文书，就Word2vec表征文本向量的不足，提出新的文本向量表征方式，并根据提取的数据特征实现了律师推荐系统。Lei等 [M Lei et al., 2017]应用不同机器学习算法，在同样裁判文书数据集进行文本自动分类，并对比了不同算法的优缺点。Feng等[Feng Y et al., 2018]基于案由与法条的共现信息，利用神经网络完成法条推荐任务。

国外最初研究人工智能法律要追溯到1987年举办的首届国际人工智能与法律会议（ICAIL），并经过历史的发展，在1991年促成了国际人工智能与法律协会（IAAIL）的成立，为促进人工智能与法律这一交叉学科的发展起了重大作用。Cardellino C等[Cardellino C et al., 2017]通过创建法律领域的实体识别模块以及分类器，提高了法律文本的信息抽取质量。García-Constantino M 等[García-Constantino M et al., 2017]研究了将已制定的文本布局应用在法律文书上，在满足CLIET环境下将有效提高法律文本的信息抽取质量。Bansal T 等[Bansal T et al., 2016]分析了主题模型对于文本推荐存在的缺点，并提出一种端到端的深度学习模型GRU进行文本特征学习，并完成文本推荐。

法条推荐作为一项基本推荐任务，本质离不开推荐系统。对于文本推荐而言，现在的主流推荐算法有以下三种：

（1）协同过滤推荐算法，该算法与内容无关，不需要分析物品内容，是基于用户的历史行为完成推荐。

（2）基于内容的推荐算法，将文本内容转为特征向量表示，基于相似度计算的算法完成推荐任务。

（3）将文本推荐转为多分类问题，将多分类的算法应用于文本推荐，诸如支持向量机、决策树算法、Bagging算法、Boosting算法以及神经网络算法均可解决上述问题。

但对于法条推荐而言，用户可能是第一次且仅有一次使用推荐系统，采用协调过滤算法是不可能实现的。而基于内容的推荐算法限制于文本特征表示的好坏与计算相似度算法，选择不当的情况下会严重影响推荐效果。所以本文采用多分类的思路进行法条推荐，优点在于利用机器学习算法相比于计算相似度，机器学习算法的泛化程度更高。并且，在特征表征上，可以通过端到端的深度网络自动学习特征，避免人工特征设计不当的情况发生。

对于使用多分类算法做文本推荐，有学者已经做了相当多贡献。Michal等[Michal Rosen-Zvi et al., 2004]以主题模型作为主要算法，额外引入作者信息作为推荐策略，在推荐学术论文时取得了良好的效果。Suchal J等[Suchal J et al., 2010]研究了利用K近邻算法实现文本推荐，并在全文搜索中取得了良好的效果。Hyung Z等[Hyung Z et al., 2014]利用PLSA分析对音乐的文本描述，来达到推荐音乐的目的，并取得了良好的效果。

## 1.3 本文研究内容与贡献

由于用户可能首次并且仅有一次使用推荐系统，用户历史行为无法获取，基于协同过滤算法难以完成。而采用基于内容的方式，推荐效果限制于文本特征提取的方法设计以及相似度计算方法。所以将法条推荐任务转为多分类问题，是本文的研究重点。而随着深度学习在人工智能领域发展日渐成熟，深度学习在法律领域也有了广泛应用。因此，本文以公开的裁判文书为数据源，着重研究了基于深度学习的法条推荐方法。

本文首先分析了法条推荐问题的本质和存在的挑战，归纳了裁判文书在结构与内容上的特点。其次，本文提出针对裁判文书进行预处理的方法，其中包括分词、去停用词、法条标准化以及将数据转为TFRecords文件。然后，本文从裁判文书引用法律法条的实际场景，即法律文书对每条法条的选择只存在两种情况，即引用与不引用，提出了多标签多分类思想，由此引出非均衡样本学习的问题。并从文本特征向量表征入手，分析了Word2vec算法与LDA主题模型在文本特征表示上的作用，并将预训练的词向量与LDA主题模型，加入端到端的深度学习模型中辅助训练，使得收敛更快速更准确。

为了验证先验知识对于法条推荐任务的重要性，即验证LDA-LSTM模型的推荐效果，主要设计了两个模型进行对比。一个是将预训练的word2vec词向量模型，用于初始化LSTM词嵌入层，加入LSTM深度网络训练，决策时直接基于隐层输出向量的平均池化结果作为特征向量，进行多标签多分类。对比实验则额外引入LDA文本向量作为先验知识，结合Attention机制，重新计算用于推荐法条的特征向量，以达到更优的推荐结果。除此之外，为了验证端到端模型自动学习特征的有效性，额外使用tfidf向量作为特征向量，通过LightGBM来进行法条推荐。

在实验阶段，本文选取六大案由的文书作为训练数据，通过一系列实验，将本文提出的基于LSTM的法条推荐、基于LDA-LSTM的法条推荐与基于LightGBM的法条推荐方法进行比较，验证了本文提出的LDA-LSTM模型的有效性，说明了先验知识对于法条推荐的重要性。最后本文分析了各个方法的优缺点，阐述了未来可能工作的方向。

## 1.4 本文的组织结构

本文的组织结构如下：

第一章 引言部分，主要介绍了论文的研究背景，国内外研究现状以及本文主要的研究内容与贡献。

第二章 为相关背景知识介绍，包括多标签多分类学习、非均衡样本学习、神经概率语言模型Word2Vec模型、LDA主题模型算法、LSTM深度网络以及Attention机制。

第三章 详细介绍了两种基于深度学习的法条推荐方法。其中主要分析了法条推荐问题以及面临的难点，根据裁判文书的数据特点提出了裁判文书的预处理方法。并详细介绍了两种法条推荐方法的训练流程以及模型结构，包括基于LSTM的法条推荐方法与基于LDA-LSTM的法条推荐方法。

第四章 实验分析部分，首先对实验所用的实验数据进行了详细介绍，然后从多个角度介绍了法条推荐模型的评估方法，接着对模型超参数设置进行了分析，最后通过一系列对比实验验证了本文提出的LDA-LSTM方法的有效性，并对不同法条推荐方法优缺点进行了分析。

第五章 总结与展望，首先总结了本文所做的研究工作，并对未来可能的工作方向做了进一步展望。

# 第二章 相关背景知识

## 2.1 多标签多分类学习

传统的有监督学习主要是单标签学习，而现实生活中，由于样本的复杂性，可挖掘的目标不止一个，往往含有多个挖掘目标。多标签多分类学习作为机器学习重要的学习方法之一，被广泛应用于文本分类、图像识别、目标检测中。有些类别上存在相互依赖，经常会同时出现在一个样本中，如何解决类标签之间的依赖性问题也是一个难点。因此，本节将主要介绍多标签多分类学习。

学习难点在于类别标签数量不确定，有些样本只有一个标签，有的样本有多个标。目前对于多标签多分类学习算法，从问题解决的角度，可以大致分为两类算法，分别是基于问题的转化方法以及基于算法适用的方法。

（1）基于问题的转化方法

当不考虑标签之间的依赖性时，可以将每一个标签看为二分类的单标签，进行one-hot编码。举例如下，假设训练样本存在四个类别，某样本类别标记为，则训练时该样本的label标记为。则需要训练4个分类器，在训练时4个分类器的label标记依次1、1、0、0，但是这种方式自动忽略了标签之间的关联信息。

（2）基于算法适用的方法

在深度学习方法中，往往可以通过巧妙设计输出层来达到多标签分类的目的。常见的方法包括在输出层的每个神经元采用sigmoid函数进行激活进行二分类，样本总体类别个数与输出层神经元个数相同。

从以上分析可知，基于问题的转化方法需要训练多个分类器，训练复杂度非常高。并且，各个分类器相互独立，失去了标签之间的关联性。而基于算法适用的方法，训练复杂度比基于问题的转化方法低，计算损失函数时由各个标签损失共同作用，一定程度上考虑标签之间的关联性。

## 2.2 非均衡样本学习

类别不平衡时训练集合中不同类别的训练样本个数差别很大，会导致学习的分类器倾向将所有样本预测为样本个数较多的类别。在现实学习任务中，经常会遇到非均衡样本的学习，比如在拆解多分类或者多标签分类问题时，即使原始问题中不同类别样本个数相当，训练子二分类器仍然可能会遇到样本不均衡的问题。因此，本节将主要介绍非均衡样本学习。

对于法条推荐而言，每个法律文书都会引用不止一条法条，由此便引出了多标签多分类学习。当不同类别的训练样本数据量相差较多时，比如负类样本个数与正类样本个数比例为100：1时，便会存在倾向于将所有标签预测为负类，然而这样的学习器没有任何价值，因为它不能准确预测出正例样本标签。

以线性分类器对样本x进行二分类为例子，再判断x为正例还是负例的时候，如果，那么将结果预测为正例，否则预测为反例。其中，y实际意义上就是指预测为正例的可能性。当训练集中正负样例个数不均衡时，令m表示为正例的个数，n为反例的个数，那么正例的观测概率为。那么如果，即预测为正例。更简单的，只需即将样本x预测为正例，这是解决非均衡样本学习常用的“再缩放”思想。

除此之外，仍有其他方法可以解决非均衡样本学习。一是去除个数较多的部分负例，称之为“欠采样”，使得正、负例样本个数接近再进行学习；二是对正类样本进行“过采样”，随机复制部分正类样本增加正例，再进行学习。欠采样在训练开销上要远小于过采样，但容易造成因为数据不足导致无法收敛到最优。需要注意的是，欠采样不是简单丢弃负例，而是采用集成学习的思路，将负例分为多个子集，训练出多个子分类器，最后综合多个子分类器结果。

## 2.3 Word2Vec模型

Word2vec模型[Le Q et al., 2014]作为一种重要的Embedding方法，有CBOW（Continuous Bag-of -Words Model）与Skip-gram（Continuous Skip-gram Model）两种模型。训练方法有基于Hierarchical Softmax和Negative Sampling两种方法，由于本文在训练时采用的是基于Hierarchical Softmax的CBOW模型，因此本节将主要介绍Hierarchical Softmax训练方法以及CBOW模型。



图2. 1 CBOW模型

如图2.1所示为CBOW模型，已知当前词的上下文的前提下预测当前词。该模型的训练方式由于没有人工标签的工作，被普遍认为是非监督学习的训练方式。基于神经网络语言模型优化时，CBOW模型优化目标通常取样本的对数似然函数如公式（2. 1）所示：

公式（2. 1）

最大化L即可找到最优解，那么现在只需要求出即可表示出对数似然函数。的计算可以通过一系列语言建模的方法，包括N-gram模型以及神经网络算法。N-gram模型假设当前词与前N个词有关，通常往往假设N=2，那么当前词只与前一个词有关。然是，该假设所能记忆的序列过短，难以处理长距离依赖的问题。如增大N，那么模型的复杂度将呈指数增长。即假设N=3，每三个词语可能的组合数将远远大于每两个词语的组合数，使得模型难以训练。因此，本文采用神经网络算法，由于不需考虑转移概率，大大降低了模型复杂度。



图2. 2 基于Hierarchical Softmax的CBOW模型

如图2.2所示，该模型是基于Hierarchical Softmax的CBOW模型。其模型结构包含三层：输入层、投影层以及输出层。以输入样本（Context(w),w）为例，假设上下文各取c个词，对于输入层，包含了Context(w)中2c个词向量；在投影层将对2c个词向量做累加求和，输出的特征向量如公式（2. 2）所示：

公式（2. 2）

传统意义上的神经网络输出层为一层神经元，而CBOW的输出层为一颗Huffman树，这棵树以词典中每个词作为叶子节点，严格按照语料库中所有词的出现次数建立的。树中每个节点都拥有两个子节点，假设词典的维度为N，则非叶子节点有N-1个。该模型中涉及的网络参数较多，包含叶子节点对应的词向量，非叶子节点对应的权重向量。

模型本质上是根据某个词的前后若干词来预测该词，是一个多分类的过程。虽然可以利用多分类的方法求出，但是语料库词典所包含词的个数很大甚至可以达到万的数量级，于是Hierarchical Softmax采用了层级二分类组合的方式计算。树中每个叶子节点都有一个向量即待训练的词向量，非叶子节点都有一个辅助权重向量，并且每个非叶子节点的权重参数不同，神经网络学习的label标签即为Huffman树的叶子节点对应的编码。

设某词w前后若干个词用表示，二叉树一共有s层，从第二层到节点w的Huffman编码为，所经过的非叶子节点对应的权重向量为，那么在上下文给定的情况下，条件概率如公式（2. 3）所示：

公式（2. 3）

其中为根据上下文得到的输出特征向量，而每个非叶子节点对应都是logistic回归问题，即标签只能为“0”或“1”。假设0表示正类，1表示负类。那么可按照公式（2. 4）计算：

公式（2. 4）

最终得到对数似然函数如公式（2. 5）所示：

公式（2. 5）

最大化对数似然函数，只需要适用梯度上升法即可。再优化过程中，叶子节点词向量和非叶子节点辅助权重向量都将作为待学习的参数不断更新，最后即可完成词向量的学习。

Skip-gram模型则是在已知词的前提下，预测上下文。基于神经网络语言模型优化时，Skip-gram模型优化目标通常取样本的对数似然函数如公式（2. 6）所示：

公式（2. 6）

只需表示出，利用梯度上升法即可训练好模型得到词向量。对于CBOW与Skip-gram模型而言，因为Skip-gram在预测的时候会进行多次预测，所以Skip-gram的训练复杂度要远远高于CBOW。但是当语料库中生僻字词较多时，由于Skip-gram要预测多个词才能更新一次词向量，采用Skip-gram模型可以让生僻字词得到充足的训练。这就相当于有多个老师在教导一个学生，那么该学生受教的程度就越高。

## 2.4 LDA主题模型算法

LDA（Latent Dirichlet Allocation）主题模型与SVD奇异值分解、PLSA等算法类似，可以用于文本语义挖掘，得到文本的主题分布。Wang C等[Wang C et al., 2011]通过主题模型完成文本的推荐，达到很好的推荐效果。而训练出主题实际上是一种抽象的词分布，无法获知该分布的具体含义。LDA主题模型作为自然语言处理重要模型之一，它被广泛应用于文本特征提取、文本分类、文本推荐中。不同于向量空间模型提取的文本向量，LDA主题模型提取的主题向量维度通常较小，并且每个维度对应着一个主题含义，含有丰富的语义信息。

LDA主题模型的发展最早追溯到Unigram Model，再到PLSA（Probabilistic Latent Semantic Analysis）模型，最后才发展到了LDA模型。LDA模型是贝叶斯学派思想下的框架，PLSA为频率学派思想框架。PLSA认为模型参数虽然未知，但始终是一个固定值。对于LDA模型而言，认为主题分布参数与词分布参数不再是固定值，而是随机变量，所以它们应该服从一定的先验分布。由于观测到的数据是呈多项分布的，所以参数与参数应该服从Dirichlet先验分布。



图2. 3 LDA概率图模型

如图2.3所示为LDA概率图模型，其文本生成过程为，为了描述方便，记表示图2.3中的：

（1）根据超参数为的Dirichlet先验分布采样K个词分布。

（2）对于过程，在生成第m篇文档的时候，根据服从超参数为的Dirichlet先验分布采样出主题分布，根据采样的主题分布生成文档第n个词的主题编号

（3）对于过程，为生成第n个词的实际过程。根据主题编号选出对应的主题下词分布，生成词 。

M篇文档对应M个独立的Dirichlet-Multinomial共轭结构；K个主题也会对应有K个独立的Dirichlet-Multinomial共轭结构。语料库W在LDA模型下的生成概率如公式（2. 7）所示：

公式（2. 7）

其中为第m篇文档采样出的主题分布，表示第m篇文档有个词。因为每个主题都可能生成词，表示对所有主题进行遍历。表示第m篇文档的第n个词，表示第m篇文档的第n个主题。

（1）学习算法

即使有了上述的联合和分布，也很难计算出似然函数具体形式，更无法得到与的估计值。因此，通常会采用近似学习算法来学习上述参数。常用的近似学习算法包括变分推断、吉布斯采样等算法。对于吉布斯采样，其目的是计算出条件概率，以便更新词语所属的主题。通过吉布斯采样算法，得到LDA模型吉布斯采样如公式（2. 8）所示：

公式（2. 8）

对于任意一个词而言，可以通过采样数据，采样出的数据将服从的计算规律，而与成正比，由此可以计算出该词所属概率最大的主题。具体流程如下：

对语料库中每个词w，随机初始化一个主题k；扫描语料数据集，对每个词w，根据吉布斯采样公式计算出该词最有可能的主题；不断利用吉布斯采样公式采样直到收敛；对采样得到的数据集合{(z, w)}，通过统计共现情况，计算出参数与参数即可

（2）推理算法

对于新的文档，计算其主题分布向量过程如下：先为文档的每个词w随机初始化一个主题；基于训练好的LDA模型主题下词的分布，根据吉布斯采样公式更新其最有可能的主题；不停使用吉布斯采样公式更新迭代，直到收敛；统计中的(z，w)对，计算出主题分布。

## 2.5 LSTM深度网络与Attention机制

近几年，基于LSTM（Long short-term memory）与Attention机制（Attention mechanism）的深度学习方法被广泛应用于文本生成、文本分类以及文本摘要等任务上。由于能较好的处理并记忆序列信息，在处理序列问题上通常不失为一种选择。

### 2.5.1 LSTM深度网络

近几年，随着计算机计算能力的提升，深度学习领域快速发展。LSTM（Long short-term memory）作为深度学习重要方法之一，由于有较长时间的信息记忆功能，被广泛应用于文本分类、文本摘要、序列数据处理以及自动问答等任务中。Sundermeyer M等人[Sundermeyer M et al., 2012]深度研究了LSTM用于语言建模的过程。Graves A等人[Graves A et al., 2013]利用双向LSTM完成了语义识别的任务。LSTM属于端到端的深度模型，不用设计特征工程，而由网络自主学习，避免了由于人工特征工程设计不当带来的问题。除此之外，LSTM作为RNN（Recurrent Neural Network）的改进网络，很好地解决了RNN训练时容易梯度消失地问题。

RNN网络中神经元一般会用到tanh函数作为激活函数，当进入tanh函数平缓区甚至饱和区，将导致梯度趋于0。对于RNN而言，在时间序列上处理长距离依赖时会遇到很大的问题，在距离较远地两节点进行误差反向传播时，会遇到雅可比矩阵多次相乘，由此带来的梯度消失问题经常导致网络训练失败。实际上，梯度爆炸也是不可避免的，但通常可以通过梯度截取解决梯度爆炸的问题。所以，找到一种改进的RNN是很必要的，在这样的背景下，LSTM被提出来。LSTM设计的巧妙之处在于通过增加输入门、遗忘门与输出门，通过门机制的作用，记录很久之前的信息，用记录的结果直接影响当前时刻的预测。具体做法是引入了单元状态c，让其保存长期状态。



图2. 4 LSTM展开后网络

如图2.4所示为LSTM展开后网络，在任意时刻，LSTM的输入有上一时刻的单元状态，上一时刻的隐层输出向量以及当前时刻的输入向量。而每一个时刻的输出有两个为当前时刻的单元状态以及隐层输出向量。



图2. 5 LSTMCell内部结构图

如图2.5所示为LSTMCell内部结构图，为上一时刻隐层的输出向量，为输入向量，主要计算包括：

（1）遗忘门

决定了上一时刻的单元状态保留的信息，遗忘门输出门向量如公式（2. 9）所示：

公式（2. 9）

（2）输入门

输入门主要有两个作用，一个是用tanh激活函数作用计算激活后用于更新的信息，其计算如公式（2. 11）所示；另一个用sigmoid函数激活充当门作用，决定需要更新的信息，其中为输入门门向量，如公式（2. 10）所示：

公式（2. 10）

公式（2. 11）

那么当前时刻的单元状态如公式（2. 12）所示：

公式（2. 12）

（3）输出门

输出门决定了有多少信息将作为向量，输入给下一个时刻。其中为输出门门向量如公式（2. 13）所示，为将要输入给下一个时刻的输出向量如公式（2. 14）所示：

公式（2. 13）

公式（2. 14）

LSTM网络包含的参数有遗忘门，输入门，输出门的权重与偏置以及输入激活的权重与偏置，即。LSTM在训练时，误差项的反向传播包括沿时间的反向传播以及沿着网络层的反向传播，通常采用BPTT（Back Propagation Through Time）算法训练网络中的参数。

LSTM通常用于处理序列信息，以文本生成任务为例，LSTM的学习过程如图2.6所示，每个时间步输入一个词，循环记忆每一步的信息，最后LSTM将预测出下一个最有可能的词。



图2. 6 LSTM文本生成

需要注意的是，图2.6所示LSTMCell是抽象的，其内部隐藏着许多神经元，包括输入门神经元，输出门神经元，输入激活神经元以及遗忘门神经元。LSTM沿着时间步展开后，每个时间步对应输入一个词的向量表示。具体地在第一个时间步，词语“我”经过Embedding后的向量输入到第一个时间步的输入神经元，经遗忘门、输入门，输入激活，输出门的作用后将单元状态c与隐层输出向量h传给下一时刻，重复上一时刻的运算。而对于当前时间步的单词预测，如果词典维度为V，首先将当前时间步的输出向量h通过全连接神经网映射到V维，再通过softmax激活函数激活，取概率最大的词即为预测结果。

### 2.5.2 Attention机制

与人类自主的注意力相似，注意力机制 (Attention mechanism) [Vaswani A et al., 2017]可以获得目标区域中的关注焦点，而后对该区域投入更多的注意力资源，来获取更多有用的信息，而忽视其他区域无用的信息。Attention机制作为重要的深度学习方法之一，在机器翻译、自动问答等序列标注问题上有广泛的应用。Luong M T等人[Luong M T et al., 2015]利用Attention机制完成自动翻译任务。Xu K等人[Xu K et al., 2015] 利用Attention机制完成根据图片生成图片描述的任务。2017年在Google发表Attention机制研究成果后，各个领域相继采用该方法对现有模型进行改进，都取得了很好的成果。目前，Attention机制已被应用于Google翻译系统中，翻译效果相比以前更是提高不少。神经机器翻译作为成功应用之一，相关研究与源代码已经发表在Tensorflow官网，由此可见Attention机制的重要性以及有效性。

通过在编码阶段引入Attention机制，对原序列数据进行加权变换，再根据变换后的结果进行预测。以机器翻译为例，将”I love China”翻译为“我爱中国”时，在解码器解码“爱”时，应该更多的关注“love”的含义，于是会赋予编码器每个词不同的权重，根据Attention机制计算出新的注意力向量，然后基于该新向量作预测。



图2. 7 基于LSTM的seq2seq模型

如图2.7所示的基于LSTM的seq2seq模型，编码器与解码器都使用LSTM模型来对序列进行学习，经编码器作用后，源序列被编码为固定向量。解码器基于该向量完成每一个时间步的解码，导致每个时间步的用于解码的向量都是链式依赖的，其效果受限于该固定长度的向量表示。除此之外，对于输入序列过长的情况，虽然LSTM有较长的记忆力，但由于链状记忆衰减，使得它对过长的序列仍然无法处理，以至于整个模型受到了序列长度限制。具体来说，编码器Encoder最后一个时间步的状态向量state将作为Decoder输入，但是状态向量state能记忆的信息毕竟有限，解码器的每一步都和以前的输入没有任何关系，只与传入的state向量有关，将限制解码器对于输入序列信息的理解能力。

为了解决上述问题，Google研究出了Attention机制。解码器端的隐层向量组成的矩阵作为Q，编码器端输出的隐层向量组成的矩阵作为K与V，计算新的注意力向量，解码器每个时间步的预测都基于不同的注意力向量完成。其计算方法如公式（2. 15）所示：

公式（2. 15）

其中Q为查询向量，比如解码器各个时间步的输入向量组成的矩阵，维度为。K为编码器的输出向量生成的矩阵，是Attention的对象，维度为。V通常和K相同，维度为。通过上述计算，将的序列Q编码为的新向量。Attention机制通过保留编码器LSTM对输入序列每一步的输出向量，然后训练一个模型对向量进行选择性的学习，并在解码的时将预测动作与之关联。



图2. 8 基于LSTM与Attention机制的seq2seq模型

如图2.8所示，编码器每个时间步接受一个词的Embedding向量，并记录每个时间步的输出向量。解码器在解码一个时间步的信息时，会将解码器上一时间步的输出向量与编码器每个时间步的输出向量计算相似度权重，再对编码器每个时间步的输出向量进行加权求和，得到上下文向量context vector。根据上下文向量以及向量算出注意力向量attention vector，通过softmax激活得出预测标签。

attention weights的计算如公式（2. 16）所示：

公式（2. 16）

context vector的计算如公式（2. 17）所示：

公式（2. 17）

attention vector即注意力向量的计算如公式（2. 18）所示：

公式（2. 18）

对于计算attention weights的score函数而言，有很多方法可以实现，较为著名的包括Luong’s multiplication方法和Bahdanau’s addition方法，计算方式如公式（2. 19）所示：

公式（2. 19）

在解码器每一个时间步进行预测时，会将输入序列转为向量序列并从中自适应的选出一个子集来解码输出。无论输入序列长短如何，解码器都是从中选出有用的子集而非全集来解码，有效解决了由于输入序列过长带来的信息记忆不充分问题。但是需要注意的是，由于需要额外计算注意力权重，上下文向量等中间结果，加入Attention机制会额外增加许多计算量，所以对计算资源有较高要求。

## 2.6 本章小结

本章介绍了与论文有关的背景知识，首先介绍了多标签学习与非均衡样本学习。其次介绍了词向量概念，并引出了神经概率语言模型Word2vec，分析了两种子模型以及训练方法。接着介绍了LDA主题模型，按照其发展历程介绍了如何对文本生成进行建模，并由此得到文本主题向量。最后，介绍了LSTM深度网络的原理以及Attention机制，分析了LSTM内部结构，LSTM在处理文本上的优缺点以及Attention机制如何解决LSTM无法解决的问题。

# 第三章 法条推荐方法设计

本章主要介绍两种基于深度学习的法条推荐方法。3.1节主要分析了法条推荐问题以及裁判文书数据集的特点。3.2节主要介绍本文提出的法条推荐方法的总流程。3.3节主要介绍裁判文书预处理方法。3.4节与3.5节分别介绍了两种法条推荐方法的流程与模型设计，分别是基于LSTM的法条推荐方法以及基于LDA-LSTM的法条推荐方法。

## 3.1 法条推荐系统总体框架

在智能化与司法公开的背景下，基于裁判文书的法条推荐技术不仅可以辅助法官判案，还能保证司法的公平公正，以及为没有法律经验的当事人提供权威的法律参考。在当下社会矛盾日渐增多的情况下，大量案件涌入法院，而法院的有限人力使得法院办公低效，本技术作为辅助工具可以为法官推荐案件可能适用的法律法条，提高其工作效率。同时，在法院司法审判的过程中，由于法官判案的主观因素，会导致“同案不同判”，通过推荐的法条与实际引用的法条作差异对比，维护司法公平公正。除此之外，利用法条推荐技术，为没有经验的当事人提供法律参考，形成最佳的诉讼策略，节约了诉讼成本。



图3. 1 法条推荐系统内部框架图

法条推荐系统的内部框架如图3.1所示，先通过裁判文书预处理模块对裁判文书数据集进行标准化和去噪，得到相对标准的数据集。基于标准的数据集合，训练法条推荐模型。当法官或者当事人输入案件基本情况后，根据预先训练好的法条推荐模型，预测出可能用于该案件的法条，并按照适用可能性大小排序输出。而实现系统则需要深入了解法条推荐问题的本质，以及法律文书数据集合的特点。接下来，本节将着重对法条推荐问题以及裁判文书数据集合特点进行分析。

### 3.1.1 法条推荐问题分析

法条推荐是在不知道案件案情与法律发条适用关系的情况下，根据用户输入的案件基本情况，为用户推荐可能适用该案件的法条。因此，能否准确的求出案件基本情况与法律法条适用关系成为了法条推荐任务的关键。

对于法条推荐问题，具体描述如下。已知以下案件以及案件引用的法条如表3.1所示，给定某案件基本情况，求出案件可能适用的法律法条。

表3. 1 案件以及引用法条

|  |  |
| --- | --- |
| 案件 | 适用法条 |
| 案件1 | 法条1，法条3 |
| 案件2 | 法条2 |
| 案件3 | 法条2，法条4 |
| 案件4 | 法条1，法条4 |
| … | … |

作为司法审判的主要载体，裁判文书记录了司法审判信息，其中包括了案件基本情况以及适用法律法条等。由于我国司法公开的战略部署，现在可以较为轻松的获取大量公开裁判文书数据。目前中国裁判文书网公开的裁判文书数量已达到4300万，这为法条推荐研究提供了很好的数据支持。如何依据已有的裁判文书数据，通过构建数学模型，为新的案件推送可能适用的法条，则是本文的研究重点。

### 3.1.2 裁判文书数据特点

针对不同的数据，往往需要构建不同的模型。为了更好的构建合适准确的模型，有必要对裁判文书的结构与写作特点进行分析。作为记录司法审判活动的主要载体，裁判文书具有以下特点：

（1）种类繁多

由于社会矛盾多样，导致裁判文书也呈现出一定多样性，按照赔偿案件、民事案件、行政案件以及刑事案件可分为四个大类。而每个大类中都有不同的案由层级，举例而言，民事案件所包含的次级案由有人格权纠纷、婚姻家庭纠纷以及物权纠纷等，而每个案由下还有更小级别的案由。具体来讲，所有案由呈一个树状分布，并且该树可以是一个多叉树。不同案由的案件，其内容表达方式上也有巨大差异。

（2）半结构化以及半口语化

训练数据中的裁判文书以XML格式存储，相关字段标签包括案件基本情况、案由、事实证据以及适用法律法条等，可以通过读取XML文书的方法来获得所需要的数据。其次，由于法律文书具有一定的权威性，不同于自然表达的语言，法律文书在写作上有一定的模式，但模式并不明显，所以只根据规则提取的方式来达到法条推荐的目标是不现实的。其次，不同部分数据的可信度不同，在选择训练数据时应该选择合适的内容，以得到较高准确率的模型。

（3）有较多的领域内专有词汇

由于时法律领域的文书写作，所以不可避免地会出现专用词汇，包括“原告”，“被告”以及“当事人”等，但这些词语对于法条推荐作用不大，因此在数据预处理时应当考虑过滤停用词以及无用专有名词。

（4）案件基本情况与适用法条紧密关联

由于法官在判案时，会着重参照案件基本情况内容，由此来给出适用地法律法条。因此判决结果常常与案件基本情况紧密联系，可以通过对案件基本情况地语义分析，来更为准确进行法条推荐。

（5）案件案由与法律法条具有一定共现性

案件案由通常标识着案件的纠纷所在，而案件的纠纷焦点往往作为法官判案时的重要参考。因此，往往案由相同的案件引用的法律法条极为相似，如果能够考虑案件的案由信息，就能大幅度缩小搜索范围。

（6）法律法条本身具有“捆绑”共现性

在统计大量文书引用法律法条的数据后，发现大部分法条共现频次很高。举个例子，法条A和法条B，由于内容上的相似性，可能经常被同时引用在一篇文书中。

由于裁判文书具有以上的特点，在文本分词时需要对文本进行停用词以及无用专有词汇过滤，并结合对案件基本情况的语义分析来进行法条推荐。

## 3.2 法条推荐系统总流程

本文的法条推荐系统主要包含两个阶段，第一个是法条推荐模型的训练阶段，第二个是根据训练好的模型进行推荐的阶段。如图3.2所示为法条推荐系统内部流程图，其中法条推荐模型的训练阶段主要包括三个步骤，具体是：



图3. 2 法条推荐系统内部流程图

（1）对裁判文书数据集进行预处理。包括从XML裁判文书中抽取案件基本情况、引用法条以及案由信息，并建立标准案由与法条全集以及法律专用停用词库。并根据标准的案由与法条全集对裁判文书进行清洗，再将数据转为Tfrecords文件等过程，其中相关预处理方法将在3.3节中详细介绍。

（2）预训练Word2vec模型以及LDA模型，并基于预训练模型来训练法条推荐模型。为了实现法条推荐，本文从“查明事实段”的语义分析层面出发，提出了两种基于深度学习的法条推荐方法。

第一种方法是基于LSTM的法条推荐方法，该方法以预训练的Word2vec模型来初始化网络Embedding层，通过LSTM分析“查明事实段”的语义信息，最后通过多标签多分类方法完成法条推荐任务，该方法于3.4节详细阐述。

第二种方法是基于LDA-LSTM的法条推荐方法，该方法仍然以预训练的Word2vec模型来初始化网络Embedding层，并引入“查明事实段”的主题向量作为先验信息，利用Attention机制计算注意力向量，并基于计算出的注意力向量进行多标签多分类完成法条推荐任务，该方法将在3.5节详细阐述。

（3）法条推荐模型评估。本文以推荐结果中预测正确的法条覆盖率作为总体评估标准，来评估法条推荐模型效果的好坏。除此之外，计算模型宏平均与微平均来进一步评估模型。

如图3.2所示，使用模型进行法条推荐时包含以下几个步骤：

（1）用户输入案件基本情况

（2）对裁判文书进行分词，去停用词以及去专有名词

（3）恢复训练好的模型，将分词处理后的数据输入法条推荐模型中计算

（4）输出案件可能适用的法条

## 3.3 数据预处理

### 3.3.1 裁判文书读取

裁判文书集合中，数据以XML形式存储，通过python插件lxml可以解析xml文档，读取相应标签中的值。由于文书中不同部分的数据采信度不同，需要选择合适的数据来训练法条推荐模型，才能达到最佳的效果。经过调研大量的裁判文书，本文最终选择了查明事实段、案由信息以及引用法律法条信息用于训练法条推荐模型。在用lxml解析XML文档时，需要提供对应标签路径path，从而可获取标签信息，各信息的路径如表3.2所示：

表3. 2 各节点path路径

|  |  |
| --- | --- |
| 信息名称 | 信息路径 |
| 案件基本情况 | /writ/QW/AJJBQK/CMSSD |
| 案由 | /writ/QW/SSJL/AY |
| 引用法律法条 | /writ/QW/CPFXGC/CUS\_FLFT\_FZ\_RY/CUS\_FLFT\_RY |

需要注意的是，当对应标签不存在时，则取上一级标签的内容。即如果/writ/QW/AJJBQK/CMSSD标签不存在时，则取/writ/QW/AJJBQK内容。其次，由于《XX诉讼法》等法条对于体现案件焦点帮助不大，所以在读取数据时去除了“诉讼法”相关的法条。另外，对包含“款”、“项”的法条，去除“款”、“项”相关的信息，只保留“条”的粒度。举例如下，法条“《中华人民共和国婚姻法》第三十二条第三款第五项”将被转化为“《中华人民共和国婚姻法》第三十二条”，其目的是避免最后获得的法条个数太多，粒度过小。裁判文书信息提取结果如图3.3所示，对文书提取了查明事实段、案由以及引用法律法条等信息。

|  |
| --- |
|  |

图3. 3 裁判文书信息提取结果示意图

需要注意的是，数据读取仅仅进行了去除“《诉讼法》”以及法条中包含的“款”、“项”内容的操作，对书写错误的法律法条以及书写错误的案由并未作任何处理。

### 3.3.2 案由和法条标准化

实际情况中，常常会遇到裁判文书法律法条书写错误的问题，以“《中华人民共和国婚姻法》”为例，由于可能书写错误，该法律名称在裁判文书中会被写作为“《中华人民共和国婚烟法》”，“《中国人民共和国婚姻法》”，“《中华人名共和国婚姻法》”等等，同时法条名称中可能包含“款”以及“项”的内容。以上书写错误以及粒度过小等错误，导致了法条类别个数太多。所以在训练法条推荐模型之前，有必要先对数据集合进行清洗。对于案由名称而言，同样存在书写错误的问题。因此，本节将详细介绍如何进行数据标准化。首先需要获取标准的法条全集与标准案由全集，具体方法为：

（1）案由标准化获取标准案由全集

与法条标准化同理，统计各个案由出现的次数。书写错误的案由出现次数相对较低，选出出现次数较多的案由作为标准化目标。

（2）法条标准化获取标准法条全集

（2.1）因为书写错误的情况比较少，在裁判文书中出现的次数应该相对较低。所以统计裁判文书中所引用的法律法条以及其出现次数，选出引用频率高的法律法条作为标准化目标。经统计，本文最终筛选出次数不低于50的法条作为法条标签全集。

（2.2）将法条中出现的数字统一转化为中文表示，比如“32”将转化为“三十二”。

（2.3）经统计发现，部分错误法条引用次数仍然较高，无法通过引用频次过滤，其主要是特殊字符字符的影响导致。比如，“最高人民法院《关于人民法院审理借贷案件的若干意见》第六条”与“最高人民法院关于人民法院审理借贷案件的若干意见》第六条”，本质上应该是同一条法条，因此本文将法条中出现的所有特殊字符均去除。

其中法条标准化前后对照效果如表3.3所示。在整个语料库中，法条引用频次的统计结果如表3.4所示。本文最终选择引用频次为50作为阈值，经法条标准化，获得标准化的法律法条全集，其中包含法条263条。对于所有案由而言，经过人工排查的方式，筛选出6个不同的案由。

表3. 3 法条标准化前后对照表

|  |  |
| --- | --- |
| 标准化前 | 标准化后 |
| 《中华人民共和国婚姻法》第三十二条第三款 | 中华人民共和国婚姻法第三十二条 |
| 《中华人名共和国婚烟法》第三十条 | 中华人民共和国婚姻法第三十条 |
| 《中华人名共和国婚烟法》第32条 | 中华人民共和国婚姻法第三十二条 |
| 最高人民法院关于人民法院审理借贷案件的若干意见》第六条 | 最高人民法院关于人民法院审理借贷案件的若干意见第六条 |

表3. 4 法条引用频次统计

|  |  |  |  |
| --- | --- | --- | --- |
| 法条最低引用次数 | 0 | 50 | 100 |
| 法条的数量 | 8387 | 263 | 122 |

得到标准法条全集与标准案由全集后，对读取的数据文件进行清洗。清洗过程为：

（1）法条清洗

对于每条记录，出现在法条全集中的法条保留，没有出现在法条全集中的法条舍弃。如果某条记录经处理后没有标准法条，则舍弃该条记录；

（2）案由清洗

对于每条记录，都应该有案由信息，如果案由名称没有出现在案由全集中时，取出编辑距离最小的标准案由作为最终案由进行更新；

### 3.3.3 分词、去停用词及法律专用停用词

中文分词是将输入的自然语言描述，利用自然语言处理的技术进行匹配计算，将源中文字符串且分为一个个单独的词。与英文分词，中文分词无法根据明显的空格进行分词，而采用专有分词算法完成分词工作。其中包括最大匹配原则、信息熵算法等。

（1）裁判文书中文分词

数据质量的好坏决定了模型预测能力，作为在裁判文书预处理中的重要环节，分词的效果直接影响到了模型的效果。目前常用的分词工具包括jieba、LTP、NLPIR以及ICTCLAS等，而jieba分词则是本文采用的分词工具。在分词前，首先利用正则表达式，将源中文字符串中的非中文符号去掉，包括数字符号以及字母，然后再利用jieba分词工具进行分词。

（2）裁判文书去除停用词

裁判文书处理本质上仍然是文本处理，因此本文下载了一份用于通用领域的停用词典，其中主要包括了“的”、“不要”等通用停用词词汇。为了获得质量更好的数据，需要尽可能的去除裁判文书中的停用词，以提高模型预测的准确率。源中文字符串在分词、去除通用停用词处理后，再去除长度小于2的词语。

（3）去除专用停用词

对于裁判文书而言，由于其半口语化的写作特点，其中出现大量的专有词汇，比如“原告”、“被告”、“法院”以及“诉称”等。然而这些词对于预测案件适用的法条作用不大，反而会提高模型训练的复杂度，影响训练结果，所以应该去除。

本文借助TFIDF算法（term frequency–inverse document frequency）中的逆文档频率IDF（inverse document frequency）思想，来实现专有停用词库的构建。逆文档频率可以看作是一个词语重要性的度量，比如对于“原告”一词，基本在每一篇裁判文书中都会出现，该词对于区分文书间的区别意义不大。对于某词语的IDF值，可以根据语料库的总文件数目除以包含该词语的文件数目，再求对数即可得到IDF值。对于词w，其IDF值为：

其中为语料库中总的文档数，为语料库中包含词w的文档个数。由上述公式可以看出，越大，词w重要性越小，IDF值也越小。所以可以通过设置一个阈值，筛选出不重要的专有停用词库。本文以5万裁判文书为数据集合，用于计算IDF值并构建专用法律停用词库。由于需要事先设定一个阈值，本文在观察了IDF计算结果后，采用的阈值为1.52，即将IDF值小于1.52的值对应的词语筛选出来作为专用停用词库。使用IDF算法来构建专有停用词库，避免了人工阅读裁判文书构建词库，减少了工作力。

对“查明事实段”内容，通过分词；再根据通用的停用词典，去除通用领域停用词；并根据本节建立的法律专用停用词库，去除专用停用词语，其结果如表3.5所示。根据表结果显示，通过去除所有停用词后，大部分作用很小的词语都被去除，而与案件案情相关的词语都被保留下来。这对减小模型训练的复杂度，以及提高模型的预测准确率都有了很大的帮助。

表3. 5 分词前后对照表

|  |  |
| --- | --- |
| 分词前 | 本院经审理，认定事实如下： 2011年2月21日，被告向原告借款30000元，并出具借条 一份，约定借款期限为2011年2月21日至2011年4月21日；同年 7月19日，被告又向原告借款20000元，并出具借条一份，约定 借款期限为2011年7月19日至2011年7月26日。上述借款到期后 ，被告未能归还借款。 |
| 分词后 | ['本院', '经', '审理', '认定', '事实', '如下', '年月日', '被告', '向', '原告', '借款', '元', '并', '出具', '借条', '一份', '约定', '借款', '期限', '为', '年月日', '至', '年月日', '同', '年月日', '被告', '又', '向', '原告', '借款', '元', '并', '出具', '借条', '一份', '约定', '借款', '期限', '为', '年月日', '至', '年月日', '上述', '借款', '到期', '后', '被告', '未能', '归还', '借款'] |
| 去除通用停用词以及专用停用词 | ['年月日', '借款', '借条', '一份', '借款', '年月日', '年月日', '年月日', '借款', '借条', '一份', '借款', '年月日', '年月日', '借款', '到期', '归还', '借款'] |

至此，有关裁判文书的读取、标准化工作、分词以及去停用词已全部完成，接下来介绍两种法条推荐方法。第一种是不加任何先验知识，基于LSTM法条推荐方法；第二种是基于LDA-LSTM法条推荐方法。

## 3.4 基于LSTM的法条推荐方法

司法审判过程中，法官往往根据案件中的查明事实段，决定该案件所适用的法律法条。而对于每条法条而言，只存在适用与不适用两种情况，那么整篇文书在法条标签全集上是一个多标签分类的问题。根据2.5节对LSTM的分析，LSTM可以解决较长时间依赖的问题，可以记录较长的序列信息。为了模拟法官在工作中的实际场景，本文提出了基于LSTM的法条推荐方法。该方法包括两个核心问题：对查明事实段的语义分析以及根据得到的语义向量进行多标签多分类。接下来将详细介绍基于LSTM法条推荐方法框架，并对流程一一分析。

### 3.4.1 具体流程图

由于裁判文书具有种类繁多，半口语化的表达方式，仅仅通过基于字符的规则提取方法显然是不现实的。因此，需要从文本语义分析出发，挖掘查明事实段的语义信息来实现法条推荐。对于序列数据的处理，LSTM深度网络是很好的选择，它不仅可以记住较长的序列信息，还能在训练过程中动态的学习特征向量。所以本文提出基于LSTM的法条推荐方法，通过对查明事实段的语义分析，将其转化为固定长度语义向量，根据该语义向量进行多标签多分类。



图3. 4 基于LSTM的法条推荐方法流程图

如图3.4所示为基于LSTM的法条推荐方法流程图，在训练一定批量数据过后，应该从验证集读取数据并验证模型，避免模型过拟合。当验证集准确率下降时，即可停止训练。主要步骤包括：

（1）按照3.3节所阐述的方法对裁判文书数据集进行预处理。需要注意的是，由于语料库中的数据有规律的排序，相同案由的案件是排序在一起的。为了更好的训练法条推荐模型，需要提前将读取的数据打乱。

（2）从Tfrecords文件读取数据，训练LSTM模型

（3）对用户输入的案件基本情况进行分词以及去除无用专有词

（4）从检查点恢复模型，将分词处理后的数据输入模型，输出推荐法条

对于训练数据，如果一次性全部读入内存，将会占用很大的内存资源。为了解决内存耗费的问题，本实验将数据转为Tfrecords文件格式。Tfrecords文件是将每条记录遵从一定的数据协议格式，写入磁盘文件。而在读取该文件时，不用记录上一次读取的位置，可以自动的读入下一个batch的数据。因此避免了将所有数据读入内存，降低了内存消耗。针对基于LSTM的法条推荐方法的模型构建与训练，将在3.4.2节将详细介绍。

### 3.4.2 模型结构及训练

本文使用LSTM模型，使用预处理后的裁判文书进行训练。为了避免过拟合与参数调优，本文将数据集以8：2的比例通过分层抽样的方法拆分为训练集Dtrain与测试集Dtest，训练集主要用于训练模型。而测试集充当了两个作用，一是及时验证模型准确率，避免过拟合；二是为了在参数调优时，为模型选择最佳参数值。由于每条数据查明事实段的长度不一，不能直接采用定长的LSTM。对于不定长的序列数据输入，通常采用的学习方法有以下两种。

（1）利用mask矩阵处理不定长序列输入

取训练集合中最长的数据，将其长度length作为LSTM在时间步上的延展。对于长度不足length的数据，在输入LSTM前通过padding方法补足为length长度。但对于LSTM每个时间步的输出向量而言，即使某时间步在padding区域内，也会有输出值。因此，需要预先记录mask矩阵（0-1矩阵），用该mask矩阵与各时间步输出向量h相乘即可得到准确的输出。

举例如下：假设LSTM的length被设置为5，某句话S经分词处理后变为[词A，词B，词C]。先将分词结果padding到10，补足的部分统一用Pad\_Id表示，则该句话变为[词A，词B，词C，Pad\_Id，Pad\_Id]，并记录mask矩阵为[1，1，1，0，0]。将S输入到LSTM神经网络后，每个时间步都会有输出向量，记作。由于句子的实际长度为3，向量与是多余的计算值，将矩阵V与mask矩阵相乘即可得到正确的结果。

（2）利用动态LSTM处理不定长序列输入

对于动态的LSTM而言，Tensorflow提供了dynamic\_rnn(cell, inputs, sequence\_length)函数以便快捷的实现变长的LSTM网络。其中的inputs参数为神经网络训练时一个batch的数据输入，设一个batch有batch\_size条数据。需要注意的是，传给参数inputs的值同样需要padding到相同的length长度。传给参数sequence\_length的值为长度为batch\_size的一维数组，数组中每个元素记录了每句话的实际长度。该函数返回值为outputs与 state。其中的state是网络最后一个时间步的状态向量，是由单元状态c与隐层状态h组成的元组（c, h）。函数在实际运行时，对于长度不足length的部分，其输出向量值outputs被设置为0，而隐层状态h将延续上一个时间步的结果不变。

举例说明：设句子S经分词处理后为S=[词A，词B，词C]，句子S经padding后长度变为5，即[词A，词B，词C，Pad\_Id，Pad\_Id]，将padding后的结果输入变长的LSTM神经网络，在第4、5时间步的输出将变为0，即。而对于最后时间步隐层状态h而言，从第三个时间步开始，隐层状态h将保持不变。因为第三个时间步的隐层状态，所以最后得到的state中隐层状态。需要注意的是，每个batch的数据必须padding成相同的长度，不同的batch的数据长度可以不同。

对于上述两方法而言，方法一需要更多的计算量。因为padding部分的序列实际上可以不用计算，而方法一计算了多余的输出向量值，最后通过mask方法将多余计算消除。为了降低模型训练复杂度，本文选择方法二来对不定长序列进行处理。



图3. 5 基于LSTM的法条推荐方法模型结构图

在实现本模型训练时，采用的是变长的LSTM网络。整个模型框架图如图3.5所示，对于模型结构而言，为了模型更快速收敛以及达到更好的学习效果，本实验在Embedding层的设计采用了动态词向量。即利用预先训练好的Word2vec模型，去初始化Embedding层的词向量矩阵。输入的查明事实段内容，分词处理后，通过Embedding将中文词语转为向量，输入变长的LSTM深度网络，得到最后一个时间步的输出向量即为。最后以Vlast作为特征向量，进行多标签多分类，实现法条推荐任务。最后在进行法条推荐时，根据最后一层激活值排序，输出前K条法条作为推荐的法条。

表3. 6 法条label经one-hot处理前后对照表

|  |  |  |
| --- | --- | --- |
| 案件 | 法条label | one-hot编码结果 |
| 案件1 | 法条1，法条3 | [1,0,1,0, … ,0] （长度为263） |
| 案件2 | 法条1，法条2，法条4 | [1,1,0,1, … ,0] （长度为263） |

本方法将法条推荐作为多标签多分类的任务，即对于每个法条而言是0-1分类问题。为了减少训练的复杂度，并尽量考虑标签关联性，进行多标签多分类时采用基于算法适用的方法。假设法条全集中法条个数为263，batch\_size为128，那么每篇裁判文书将按照如表3.6所示的方法进行one-hot编码。由于每篇裁判文书引用的法条数量远远小于标准法条全集总数，每个batch的label经one-hot编码后形成[128, 263]的矩阵，矩阵中“0”与“1”的数量严重不平衡。根据2.2节对非均衡样本学习的介绍，本实验采用“再缩放”的思想来解决不均衡问题，主要通过设置一个正负样本权重pos\_weight来实现。该参数是一个超参数，需要通过在验证集合上的评估来选择最佳参数，具体选择方法将在4.4节中详细介绍。

### 3.4.3 代码实现

对于LSTM的法条推荐方法，模型主要创建代码如图3.6所示。对于整个模型，首先创建模型的占位符placeholder输入，包括self.inputs、self.targets\_y、self.seq\_lens、self.dropout。其中的inputs需要提前转为字典中词对应的id，经模型的Embedding层转为向量后，输入变长循环神经网络dynamic\_rnn中，得到每个时间步的输出向量all\_outputs以及网络最后的状态向量state。由于每个序列的长度不同，需要消除长度不同带来的影响得到outputs向量。在计算loss的时候，需要得到与labels相同维度的logits，所以需要将outputs向量通过全连接神经网络映射到与labels相同的维度，得到logits。logits中每个元素代表了法条被推荐的概率大小，在进行法条推荐时，可以根据logits向量中各个元素的大小，选择前12条进行推荐。最后tf.trainable\_variables()得到网络中所有需要更新的参数，优化器选择为AdamOptimizer，根据计算得到的loss对网络进行更新即可。

|  |
| --- |
| def \_\_init\_\_(self,FLAGS,embedding\_matrix):  设置placeholder占位符，有self.inputs、self.targets\_y、self.seq\_lens、self.dropout  self.global\_step = tf.Variable(0, trainable=False)  with tf.device('/cpu:0'),tf.name\_scope("embedding"):  初始化Embeding矩阵self.embedding  inputs=tf.nn.embedding\_lookup(self.embedding,self.inputs\_X)  stacked\_cell=tf.nn.rnn\_cell.MultiRNNCell([self.rnn\_cell(FLAGS,self.dropout) for \_ in range(FLAGS.num\_layers)])  all\_outputs,state=tf.nn.dynamic\_rnn(initial\_state=initial\_state,cell=stacked\_cell,inputs=inputs,sequence\_length=self.seq\_lens,dtype=tf.float32)  outputs=tf.reduce\_sum(all\_outputs,1)/self.seq\_lens[:,None]  #将outputs经dense layer处理，得到logits  self.predict=tf.nn.top\_k(logits,12,sorted=True)  self.loss =  tf.reduce\_mean(tf.nn.weighted\_cross\_entropy\_with\_logits(self.targets\_y,logits,FLAGS.pos\_weight))  self.lr = tf.Variable(0.0, trainable=False)  trainable\_vars=tf.trainable\_variables()  optimizer=tf.train.AdamOptimizer(self.lr)  优化器对网络参数进行更新 |

图3. 6 基于LSTM的法条推荐方法模型构建代码

模型的训练程序如图3.7所示，首先设置GPU的使用方式，比如独占或是动态分配显存，本方法选择动态分配的方式。之后通过tf.Graph()建立计算图，在计算图中创建模型结构。需要注意的时，实验是通过读取Tfrecords文件获取训练数据与测试数据的，并且为了减小训练时内存的消耗，通过inputs()函数创建了两个数据读取线程，分别时读训练数据线程与读测试数据线程。线程将数据读入相关队列，而训练主线程则从队列中获取数据并用于训练。训练一定量数据后，需要执行模型验证，以避免模型过拟合。

|  |
| --- |
| def train(vocab\_dict):  设置GPU的适用方式  embedding\_matrix = get\_EmbeddingMatrix(vocab\_dict)  #根据Word2vec得到embedding\_matrix，用于初始化深度网络的Embedding层  with tf.Graph().as\_default(), tf.Session(config=gpuConfig) as sess:  train\_fact, train\_laws =  inputs(FLAGS.input\_traindata, FLAGS.batch\_size,FLAGS.num\_classes)  valid\_fact,valid\_laws=  inputs(FLAGS.input\_validdata,FLAGS.batch\_size,FLAGS.num\_classes)  model =create\_model(sess,FLAGS,embedding\_matrix)  coord=tf.train.Coordinator()  threads=tf.train.start\_queue\_runners(coord=coord)  从训练数据存储线程中获取数据训练模型，每训练一定数量数据再从测试数据线程中读取数据并验证模型，避免过拟合 |

图3. 7 基于LSTM的法条推荐方法模型训练代码

## 3.5 基于LDA-LSTM的法条推荐方法

司法审判过程中，如果以文书主题作为约束，那么引用法条必定与主题有较强的关联关系。所以，本实验额外加入主题向量作为先验知识，仍基于LSTM来训练法条推荐模型。该方法核心思想通过LDA主题模型计算出查明事实段的主题向量，将主题向量作为先验知识，输入到LSTM深度网络中训练。其中，结合先验知识的方法是需要解决的核心问题，接下来将详细介绍基于LDA-LSTM的法条推荐方法。

### 3.5.1 具体流程图

由于加入先验知识可以提供更多信息，对法条推荐任务有着重大的意义。所以本实验在训练LSTM模型的时候，额外引入了LDA主题向量。其具体流程如图3.8所示，其主要步骤包括：



图3. 8 基于LDA-LSTM的法条推荐方法流程图

（1）按照3.3节所阐述的方法对裁判文书数据集进行预处理，得到标准化数据的Tfrecords文件。同样地，由于语料库中的数据有规律的排序，模型训练前需要将数据乱序处理。

（2）从Tfrecords文件读取数据，训练基于LDA-LSTM的法条推荐模型。再训练阶段需要定期对验证集中数据进行验证，以避免过拟合。

（3）对用户输入的案件基本情况进行分词以及去除专用停用词。

（4）从检查点文件恢复模型，将分词结果输入模型，输出推荐法条。

接下来3.5.2节将详细介绍本实验的模型结构以及训练方法。

### 3.5.2 模型结构及训练

同样以8：2的比例，按照分层抽样的方法将数据拆分为训练集Dtrain与测试集Dtest，所用方法和目的与3.4节相同。为了减少训练时所消耗的内存，仍然预先将训练集Dtrain与测试集Dtest转为Tfrecords的文件格式，同时实验对于处理不定长输入序列仍然采用动态LSTM的方法。该实验的关键在于设计一种有效的方法将LDA先验知识引入LSTM深度网络中，接下来将详细介绍。模型结构如图3.9所示，为了更快的收敛以及更高的准确率，Embedding层采用预训练好的Word2vec模型来初始化。首先输入案件的基本情况，分词结果经过Embedding层转为向量表示后，再输入到变长深度网络LSTM中训练，接下来详细介绍LDA主题向量的引入。



图3. 9 基于LDA-LSTM的法条推荐方法模型结构图

记主题向量为LDA\_vector，编码器每个时间步的隐层输出向量，并假设编码器一共延展S个时间步。案件基本情况通过LDA模型作用，得到主题向量LDA\_vector。将主题向量通过一层全连接神经网络映射到深度网络的学习空间，得到向量querynew。将querynew作为Attention机制中的query向量，组成的矩阵作为K与V，根据2.5.2节介绍的Attention机制，计算出注意力权重attention weights如公式（3. 1）所示：

公式（3. 1）

其中表示编码器第s个时间步的注意力权重。然后计算出注意力向量attention vector如公式（3. 2）所示：

公式（3. 2）

在得到注意力向量attention vector后，基于该向量进行多标签多分类。与3.4.2节相同，该过程是一个非均衡情况下的多标签多分类学习，需要调整正负样本的权重。最后在进行法条推荐时，根据最后一层激活值排序，输出前K条法条作为推荐的法条。

需要注意的是，如图3.9所示将LDA主题向量映射到LSTM深度网络空间后，再计算注意力向量。其理由如下：该模型结构存在两个子空间，LSTM深度网络空间与LDA主题向量空间，记LSTM深度网络空间为“空间1”，记LDA主题向量空间为“空间2”。首先，如果“空间1”和“空间2”的维度不同，是不能计算注意力向量的。其次，即使“空间1”和“空间2”的维度相同，每个维度所代表的含义也不同，进行attention计算是没有意义的。而LDA只是作为预训练的模型，实际训练的主要在学习LSTM深度网络空间。所以需要将主题向量通过一层全连接神经网络映射到LSTM深度网络空间，再计算注意力向量。

### 3.5.3 代码实现

如图3.10所示为Attention主要代码实现，为了将LDA主题向量映射到LSTM深度网络空间，通过矩阵变化得到映射后的向量u\_omega，将LSTM每个时间步的隐层输出向量与u\_omega进行attention计算，相关原理参考2.5.2节，得到LSTM每个时间步的注意力权重。通过每个时间步注意力权重与对应隐层向量可以计算出注意力向量，最后基于注意力向量进行多标签多分类。

|  |
| --- |
| import tensorflow as tf  def attention(inputs,attention\_size,topic\_vector,time\_major=False):  inputs=tf.array\_ops.transpose(inputs,[1,0,2])  batch\_size=inputs.shape[0].value  hidden\_size=inputs.shape[2].value  with tf.variable\_scope("attention\_W"):  w\_omega=tf.get\_variable("w\_omega",[attention\_size,hidden\_size],initializer=tf.glorot\_normal\_initializer())  u\_omega=tf.tensordot(topic\_vector,w\_omega,axes=1)  u\_omega=tf.reshape(u\_omega,[batch\_size,-1])  vu=tf.matmul(inputs,tf.expand\_dims(u\_omega,-1))  vu=tf.reshape(tf.squeeze(vu),[batch\_size,-1])  alphas=tf.nn.softmax(vu,name='alphas')  output=tf.reduce\_sum(inputs\*tf.expand\_dims(alphas,-1),1)  return output |

图3. 10 Attention代码实现

对于模型训练，其过程与基于LSTM的法条推荐方法类似。首先建立两个数据线程，训练主程序从线程中获取数据并进行验证。不同的是，基于LDA-LSTM的法条推荐方法需要额外通过LDA主题模型，计算出LDA主题向量，并通过Attention机制输入到模型中计算，其需要的计算资源将大大增加。因此，该方法在GPU使用的设置上，选择独占GPU，避免显存溢出。

## 3.6 本章小结

本章详细介绍了基于深度学习的法条推荐方法。在3.1节中对法条推荐问题以及裁判文书特点进行了分析。3.2节介绍法条推荐系统的总流程。在3.3节中介绍了裁判文书预处理方法，其中包括了信息读取、数据标准化以及分词去停用词等。在3.4节与3.5节分别介绍了两种基于深度学习的法条推荐方法，分别是基于LSTM的法条推荐方法与基于LDA-LSTM的法条推荐方法，并对两种方法的流程以及模型框架进行了详细介绍。

# 第四章 实验分析

本章主要描述基于深度学习的法条推荐方法的实验过程与结果分析。4.1节主要介绍裁判文书数据集；4.2节主要介绍法条推荐模型的多项评估标准；4.3节主要介绍预训练LDA主题模型与Word2vec模型的方法；4.4节对法条推荐模型的超参数进行介绍与分析；4.5节展示了多组对比实验，并对实验结果进行分析；4.6节对两种法条推荐方法的优缺点进行了分析。

## 4.1 数据集

裁判文书种类繁多，为了能泛化法条推荐系统功能，本文选取了以下六个案由的文书，分别是：离婚纠纷、不当得利纠纷、劳动合同纠纷、承揽合同纠纷、机动车交通事故责任纠纷以及民间借贷纠纷。通过法条标准化后，过滤引用频次小于50的法条后，裁判文书数据集共引用263条不同的法条。各个案由所包括的文书以及引用法条如表4.1所示：

表4. 1 裁判文书数据集统计表

|  |  |  |  |
| --- | --- | --- | --- |
| 案由名称 | 裁判文书总数 | 平均引用法条数 | 标准引用法条总数 |
| 劳动合同纠纷 | 10521 | 4.3 | 90 |
| 民间借贷纠纷 | 11629 | 4.9 | 42 |
| 承揽合同纠纷 | 12153 | 4.1 | 52 |
| 离婚纠纷 | 12625 | 3.2 | 19 |
| 机动车交通事故  责任纠纷 | 12287 | 10.1 | 80 |
| 不当得利纠纷 | 9191 | 2.7 | 24 |

根据每个案由下所有裁判文书引用的法条数目N，其中包括不规范法条，以及该案由下的裁判文书数n，计算得到平均引用法条数如公式（4. 1）所示：

公式（4. 1）

另外，统计标准引用法条总数时，按照3.3.2节所述，过滤掉引用频次小于50、将数字转为中文、去除特殊字符等步骤得到。其次，因为“诉讼”法相关法条与案件基本情况描述联系不大，所以在统计时去除了“诉讼法”相关法条。需要注意的是，由于各案由间引用的法条会有重合，导致表4.1中六个案由标准引用法条总数和为307263。

对于此数据集中的文书，通过XML文档读取技术获得每一篇文书的唯一标识、查明事实段、案由以及引用法律法条等信息，将这些信息作为训练语料训练法条推荐模型。接下来介绍对上述数据集的划分，以分层抽样的方式切分为训练集与测试集。

为了能较好的切分数据集以及评估模型性能，本文将数据集合按照8:2的比例切分为训练集和测试集。需要注意的是，划分时按照分层抽样的原理进行切分，使得每种案由的文书在训练集和测试集中的比例都相当。其中训练集用于训练法条推荐模型，而测试集用于评估法条推荐模型以及用于模型参数调优。数据切分按照表4.2所示进行。

表4. 2 切分训练集与测试集对照表

|  |  |  |  |
| --- | --- | --- | --- |
| 案由 | 文书总数 | 训练集 | 测试集 |
| 离婚纠纷 | 12625 | 10100 | 2525 |
| 民间借贷纠纷 | 11629 | 9300 | 2329 |
| 承揽合同纠纷 | 12153 | 9720 | 2433 |
| 不当得利纠纷 | 9191 | 7350 | 1841 |
| 机动车交通事故  责任纠纷 | 12287 | 9830 | 2457 |
| 劳动合同纠纷 | 10521 | 8145 | 2376 |

## 4.2 评估标准

经统计，所有文书的平均引用法条次数为4.8次，本文将从推荐条进行评估。对于推荐系统而言，通常采用召回率（Recall）与准确率(Precision)作为评估标准。以二分类为例，其混淆矩阵如表4.3所示，其中TP为预测结果为“正”并且真实标记也为“正”的样本数量；FP为预测结果为“正”但真实标记为“负”的样本的数量；FN为预测结果为“负”但真实标记为“正”的样本数量；TN为预测结果为“负”且真实标记也为“负”的数量。

表4. 3 混淆矩阵

|  |  |  |
| --- | --- | --- |
| 预测  真实 | 正 | 负 |
| 正 | TP | FN |
| 负 | FP | TN |

准确率如公式（4. 2）所示：

公式（4. 2）

召回率如公式（4. 3）所示：

公式（4. 3）

准确率体现了所有预测结果中实际预测正确的比例，召回率体现了所有真实结果中预测正确的比例。由于本文将法条推荐当作多分类问题处理，所以可用多分类问题的评估方法来评估法条推荐模型。对于多分类任务，可以转化为用二分类问题评估。假设进行n分类任务，采用一对多方式可以拆分为n个二分类评估，如表4.4所示。

表4. 4 多分类问题分解为二分类对照表

|  |  |  |
| --- | --- | --- |
| 二分类器 | 正类类别 | 负类类别 |
| 第1个 | 1 | 2, 3, …, n |
| 第2个 | 2 | 1, 3, …, n |
| … | … | … |
| 第n个 | N | 1, 2, …, n-1 |

每个二分类都有对应的混淆矩阵，一共有n个混淆矩阵。假设对于每个混淆矩阵计算出的准确率与召回率分别为（p1, r1）（p2, r2）,…,（px, rx），其中，那么宏查全率、宏查准率与宏F值如公式（4. 4）所示：

公式（4. 4）

其次，如果将n个混淆矩阵对应元素求平均值得到，那么微查全率、微查准率与微F值如公式（4. 5）所示：

公式（4. 5）

与的平均值如公式（4. 6）所示：

公式（4. 6）

以上评估方式均是从法条本身的角度来进行评估，为了充分评估对于测试集合每篇裁判文书的预测推荐状况，本文还以推荐法条结果中，正确法条的覆盖率进行评估，记正确法条的覆盖率为。在法条推荐任务中，推荐的法条能否较多的覆盖裁判文书的真实法条，是需要关注的焦点。所以，本文还采用覆盖率作为法条推荐模型的评估标准来对模型进行评估。即对一篇法律文书，真实引用了3条法条[A, B, C]，推荐法条为[A, B, D, E, F]，那么覆盖率为2/3。因此，对于有N条数据的测试集而言，其平均覆盖率如公式（4. 7）所示：

公式（4. 7）

其中为第i篇裁判文书推荐结果中正确的法条数目，为第i篇裁判文书中真实引用的法条数目。其次，为了探讨推荐法条数目K对模型的影响，本文提出占有率，其计算方法如公式（4. 8）所示：

公式（4. 8）

其中=K，而对法条推荐数目K的总体评估如公式（4. 9）所示：

公式（4. 9）

根据本节的描述，等都可以进行法条推荐模型的评估，具体评估方式根据不同的实验会有不同。

## 4.3 预训练LDA与Word2vec模型

本文法条推荐方法用到两个预训练模型，一个是LDA主题模型，一个是Word2vec词向量模型。其中Word2Vec模型用于初始化模型的Embedding层，LDA主题模型用于计算裁判文书的先验主题向量，接下来将对两个模型的训练过程进行详细的介绍。

### 4.3.1 预训练Word2vec词向量模型

Word2vec将作为预训练模型，初始化深度网络中的Embedding层词向量矩阵。根据2.3节介绍，Word2vec采用无监督学习方法，以语料库中所有裁判文书作为训练数据，学习分布式语义空间并得到每个词的词向量。其训练程序如图4.1所示，模型训练采用Gensim库，通过Word2vec()函数便可训练并建立word2vec模型。其中sentences参数为经过分词处理的语料库；size参数定义了词向量维度的大小；window参数定义了当前词与预测词在一个句子中的最大距离，即定义了上下文；min\_count参数定义了词语最少出现的次数，出现次数低于min\_count的词语将被字典舍弃。

|  |
| --- |
| def train\_word2vec(sentensces):  model = Word2Vec(sentences=sentensces,size=150,window=5,min\_count=3,work  ers=2)  model.save("graduate/w2v\_model") |

图4. 1 Word2vec模型的训练

如图4.2所示为当输入“离婚”时，输出前10个最相似词的测试结果。从图中可以看出，Word2vec模型能够找到“结婚”、“复婚”以及“分手”等一系列意思相近的词，很好的挖掘了法律裁判文书的内在语义关系。

|  |
| --- |
|  |

图4. 2 Word2vec模型测试结果

### 4.3.2 预训练LDA主题模型

根据2.4节介绍，LDA主题模型主要通过概率建模，以语料库中所有裁判文书作为训练数据，得到每篇裁判文书的主题分布以及每个主题下的词分布。对于模型而言，主题数K是一个超参数。其训练程序如图4.3所示，

|  |
| --- |
| def train\_model(input\_path,model\_path,dict\_path):  alltext=get\_context(input\_path)  train\_word=cut\_text(alltext)  dictionary=corpora.Dictionary(train\_word) dictionary.save(dict\_path)  corpus=[dictionary.doc2bow(text) for text in train\_word]  lda=LdaModel(corpus=corpus,id2word=dictionary,num\_topics=256,passes=2)  lda.save(model\_path) |

图4. 3 LDA主题模型的训练

其中get\_context()函数从语料库读取裁判文书的“查明事实段”内容，形成列表alltext。经过分词后，根据建立的词典dictionary将alltext的所有词语转为词语在字典中的词序，得到corpus。最后训练LDA主题模型，LDA训练采用Gensim库。其相关参数设置以及参数意义如表4.5所示，由于深度网络中隐层神经元数量设置为256，为了使得LDA学习的语义信息与深度网络大致相当，将主题数量设置为256。

表4. 5 LDA参数设置

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数值 | 参数意义 |
| num\_topics | 256 | 主题的数量，即将文本映射为256维的向量 |
| Passes | 2 | 对整个语料库遍历2次 |

LDA训练后每个主题的词分布如图4.4所示，这里仅展示了部分主题，一共有256个主题。其中每个主题都是抽象的，代表的是一系列词的分布，而词的分布构成了主题的含义。LDA得到文本主题向量的过程，也就是为每个主题赋予不同的权重，生成一个主题向量，即将每一篇文本映射为256维的向量。

|  |
| --- |
|  |

图4. 4 LDA主题下词分布示意图

## 4.4 法条推荐模型的超参数设置

对于两个法条推荐模型而言，除了待学习的网络参数，还有一系列的超参数设置。而超参数的设置将对模型的推荐结果产生极大的影响，所以本节将对两个法条推荐模型的超参数进行详细介绍与分析，分别是基于LSTM的法条推荐模型与基于LDA-LSTM的法条推荐模型。

### 4.4.1 基于LSTM的法条推荐方法超参数设置

该方法包含的超参数、参数值如表4.6所示，其中每个参数对应的意义为：

表4. 6 基于LSTM的法条推荐方法超参数设置

|  |  |
| --- | --- |
| learning\_rate | 0.0001 |
| max\_gradient\_norm | 5.0 |
| batch\_size | 128 |
| num\_hidden\_units | 256 |
| embedding\_size | 150 |
| num\_layers | 2 |
| dropout | 0.6 |
| en\_vocab\_size | 100000 |
| num\_classes | 263 |
| valid\_step | 30 |
| valid\_num\_batch | 53 |
| pos\_weight | 300 |
| max\_time\_step\_size | 1000 |

learning\_rate：代表深度网络的学习速率。

max\_gradient\_norm: 代表网络的梯度截断阈值，该参数主要用于避免梯度爆炸的问题。

batch\_size：代表每个batch中数据的数目，该参数过大会导致网络学习变慢或者显存溢出，过小会导致网络陷入局部最优。

num\_hidden\_units：代表LSTM深度网络中循环层神经元的数目。

embedding\_size：代表词嵌入的维度，表示将词语转为固定维度的向量

num\_layers：代表LSTM网络的层数。

dropout：代表对输出神经元或者输入神经元进行dropout正则，本实验是对输出向量进行dropout处理，用于防止模型过拟合。

en\_vocab\_size: 代表LSTM训练词典的维度。其中包括了UNK\_ID以及PAD\_ID，对于句子padding的部分，统一用PAD\_ID代替，而如果词语不在词典中，统一映射为UNK\_ID。

num\_classes：代表类的个数，法条推荐任务中即为法条全集中法条的总个数。

valid\_step: 训练阶段每经过多少步在验证集中进行验证，该参数有利于避免模型过拟合。

valid\_num\_batch: 每次验证时需要运算的batch的批量数，由这些数据来计算验证集上的准确率。

pos\_weight: 非均衡标签下样本权重调节参数。

max\_time\_step\_size: LSTM时间步延展的最大值，即每个batch的数据长度如果超过该值，按照该值截断。

本实验在模型训练时，将测试集充当验证集使用。接下来着重对pos\_weight参数以及max\_time\_step\_size参数值的选取进行分析，其他参数值都是神经网络中通常设置的参数值。

（1）max\_time\_step\_size参数的分析

该参数代表了LSTM允许的最大时间步数量。当没有对该参数进行限制时，LSTM的最大长度可能过大，导致训练时显存溢出。当该参数设置过小时，LSTM时间步延展过小，导致深度网络不能很好的学习到序列信息，模型预测能力较差。所以，在保证显存没有溢出的情况下，尽量将该参数设置较大。其中，实验所用的GPU显存为11172MB，通过实验对比，最后选取的参数值为1000，即最大支持分词后长度为1000的文本。实验结果如表4.7所示，当对max\_time\_step\_size不进行约束时，显存将溢出。而将max\_time\_step\_size设置为1000时，能最大效率的利用GPU进行计算。

表4. 7 max\_time\_step\_size与显存占用关系表

|  |  |
| --- | --- |
| max\_time\_step\_size | 最大显存占用 |
| 500 | 4095MB |
| 1000 | 10013MB |
| 不约束 | 显存溢出 |

（2）pos\_weight参数的分析

在多标签多分类时，假设batch\_size设置为128，num\_classes为263。每条数据的标签经过one-hot编码后，标签矩阵维度为[128, 263]，矩阵中的元素均为“0”或“1”。根据3.4.2节，通过调整正负样本权重可以解决非均衡样本的学习。所以通过设置不同pos\_weight值，训练了多个模型，在测试集上的准确率如图4.5所示，法条推荐模型评估方式采用，其中法条推荐个数K=12。从图中可以看出，随着pos\_weight从50上升到300，三项评估指标的值都在持续增长，而当pos\_weight超过300后，有轻微下降，而基本保持平稳。分析其原因，主要是因为[128, 263]标签矩阵中，元素“0”数量大概是元素“1”的300倍。因此，本实验选取的最佳pos\_weight值为300。

图4. 5 参数pos\_weight对模型的影响

### 4.4.2 基于LDA-LSTM的法条推荐方法超参数设置

该实验设置的相关超参数如表4.8所示，参数设置与基于LSTM的法条推荐模型基本相同，其中max\_time\_step\_size设置为1000时，显存占用为10223MB。由于基于LDA-LSTM的法条推荐方法模型结构较为复杂，所以会占用更大的显存。pos\_weight只与标签矩阵的非均衡因素有关，由于该实验训练数据与4.4.1节相同，因此该实验仍设置pos\_weight为300。

表4. 8 基于LDA-LSTM的法条推荐方法超参数设置

|  |  |
| --- | --- |
| learning\_rate | 0.0001 |
| max\_gradient\_norm | 5.0 |
| batch\_size | 128 |
| num\_hidden\_units | 256 |
| embedding\_size | 150 |
| num\_layers | 2 |
| Dropout | 0.6 |
| en\_vocab\_size | 100000 |
| num\_classes | 263 |
| valid\_step | 30 |
| valid\_num\_batch | 53 |
| pos\_weight | 300 |
| max\_time\_step\_size | 1000 |

对于LDA先验信息，可以通过Attention机制，将LSTM每个时间步输出的向量组成的矩阵作为K与V，LDA主题向量作为query向量，计算注意力向量，相关计算原理细节参考2.5.2节。该模型中，LSTM隐层神经元维度为256，而LDA主题向量的维度为256，采用的乘法注意力。根据3.5.2节介绍，对于LSTM输出的隐层向量与LDA主题向量而言，虽然维度大小一致，由于每个维度对应的意义不同，不能直接作向量内积，需要转换到同样的语义空间。那么有以下两个方法：

（1）将LSTM的输出向量通过一层全连接神经网络转换到LDA主题空间，即统一映射到LDA主题空间。

（2）将LDA主题向量通过一层全连接神经网络转换到LSTM语义空间，即统一映射到LSTM主题空间。

通过实验对比上述两种方法，实验结果如图4.6所示，法条推荐个数K=12，由实验结果可以看出，方法（1）的Flaw、CoverLaw与EvalLaw值都会比方法2的对应值低。经分析，是因为整个模型学习都是在深度网络空间中学习的，例如：深度神经网络的权重参数是对应在LSTM空间的。但是LDA主题模型只是作为预训练的先验知识模型，整个神经网络所学习的权重网络参数并不是在LDA空间中学习的，所以将LSTM的隐层输出向量转化到LDA空间会降低模型的预测准确率。

图4. 6 计算Attention的空间对模型的影响

## 4.5 对比实验

在本小节中，将分别设计一系列对比实验，来验证本文提出的法条推荐方法的效果，同时对实验结果进行详细分析。为了便于实验描述，使用LSTM代表基于LSTM的法条推荐方法，使用LDA-LSTM代表基于LDA-LSTM的法条推荐方法。在实现LSTM与LDA-LSTM模型时，均采用Google开源的Tensorflow框架设计模型并进行实验。

为了验证深度学习方法的有效性，本文将额外使用集成学习方法LightGBM[Ke G et al., 2017]作为对比。对于LightGBM而言，采用每篇文书的tfidf向量作为特征向量，利用LightGBM进行实验，而LightGBM实验所采用的是微软的开源算法库。

为了关注在测试集合中每篇文书的法条推荐效果，以下实验主要采用对实验进行评估。其中4.5.1节研究标准化标签集中法条总数对推荐效果的影响；4.5.2节研究法条推荐个数K对推荐效果的影响；4.5.3节研究预训练词向量模型对LSTM以及LDA-LSTM的影响；4.5.4节对比了LSTM、LDA-LSTM以及LightGBM进行法条推荐的效果。

### 4.5.1 标准化标签集中法条总数对推荐效果的影响

研究问题（1）：由于将法条推荐作为多标签多分类问题处理，探讨标签的总个数对法条推荐效果产生的影响。

记标准化标签集中法条总数为N，根据3.3.2节，引用次数大于0的法条有8387条，引用次数大于50的法条有263条，引用次数大于100的法条有122条。实验采用评估标准为测试集合上的CoverLaw值。实验设置K=12，实验结果如图4.7所示，在选择将引用频次大于0的法条作为全集时，此时CoverLaw值很低，而选择引用频次大于50的法条作为全集时，CoverLaw值会有很大提升，提升将近0.1。纠其原因，是因为引用频次大于0的法条中，有大量的不规范法条，数据质量太差导致评估值太低。对于最低引用频次为50和100的而言，由于法条全集基本清除完不规范法条，CoverLaw值的提升较缓，甚至趋于平稳，说明了法条标准化的重要性。为了让模型更准确，同时让模型适用于更多法条，最低引用频次为50是比较好的选择值。

图4. 7标签总数N对法条推荐效果的影响

### 4.5.2 法条推荐个数K对推荐效果的影响

研究问题（2）：法条推荐个数K值对法条推荐效果的影响

实验所用法条标签全集中有263条不同法条，即引用频次不低于50的法条。通过设置K=6、12、18进行三组实验，其中LSTM与LDA-LSTM均采用了预训练词向量模型来初始化Embedding层。实验所用评估标准为法条覆盖率CoverLaw值以及EvalLaw值。

首先，CoverLaw的实验结果如图4.8所示，从图中可以看出随着K的增大，CoverLaw评估值持续上升。而从K=6到K=12模型CoverLaw值的提升要大于从K=12到K=18的阶段。比如对于LDA-LSTM而言，从K=6到K=12，CoverLaw值增大了0.1；而从K=12到K=18，CoverLaw值增大了0.03，其他模型亦是如此。分析其原因，本文认为推荐的法条个数K越大，那么推荐出正确法条的可能越大，所以CoverLaw值会持续上升。而当推荐法条个数K抵达一定大小时，由于模型自身预测能力并不是随着K的增大线性提升，所以当K超过一定值时，CoverLaw值的提升变缓。形象的说，当前模型在某评估值较高时，继续提高模型效果将变得很难。

图4. 8 K值对模型CoverLaw的影响

实际情况下，并不是K值越大越好。即使K值越大，模型CoverLaw值越高，但推荐的结果中正确的法条占比会相对降低。对应到实际情况下，推荐的法条个数越多，包含正确法条的可能性越大，但是要从推荐结果中找到正确的法条所耗时间与工作量越大。为了解决上述问题，根据4.2节介绍，引入EvalLaw对模型进行评估，实验结果如图4.9所示。随着K增大，EvalLaw值并非持续增长。对于LSTM、LDA-LSTM与LightGBM而言，从K=12到K=18模型的EvalLaw值都有轻微下降，比如LDA-LSTM从0.64降到0.60。纠其原因，是因为法条推荐个数K越大，虽然CoverLaw升高，但OwnLaw降低，使得EvalLaw降低。所以根据实验结果，为了使的正确法条的覆盖率CoverLaw较高，同时使EvalLaw不至于太低，选择K=12作为最佳值。

图4. 9 K值对模型EvalLaw的影响

### 4.5.3 预训练词向量对LSTM、LDA-LSTM的影响

研究问题（3）：预训练词向量对LSTM与LDA-LSTM的影响

实验所用法条标签全集中有263条不同法条，即引用频次不低于50的法条，并且设定K=12，实验所用评估标准为法条覆盖率CoverLaw值。其次，为了评估模型收敛的快慢，从开始训练直到模型loss降低到0.80时，模型所使用的batch数量记为num\_batch。

预训练词向量对CoverLaw的影响如表4.9所示，加入预训练词向量情况下，LDA-LSTM的CoverLaw值有轻微提高，而LSTM的评估值基本不变。本文认为，这是因为LSTM在训练深度网络的时候，会根据语料库动态更新词向量，即使没有预训练词向量初始化的情况下，LSTM网络也能自发学习并更新词向量。

表4. 9 预训练词向量对CoverLaw的影响

|  |  |  |
| --- | --- | --- |
| 模型 | 输入预训练词向量 | 不输入预训练词向量 |
| LSTM | 0.84 | 0.84 |
| LDA-LSTM | 0.88 | 0.87 |

预训练词向量对num\_batch的影响如表4.10所示，从表中可以看出，加入预训练词向量情况下，当loss降低到0.80所需要遍历的数据大大降低。比如对于LSTM而言，没加入预训练词向量需要遍历1032个batch的数据，而加入预训练词向量只需遍历612个batch的数据，LDA-LSTM与此类似。但是由于LDA-LSTM模型较为复杂，所以需要比LSTM更多的数据来训练模型。

表4. 10 预训练词向量对num\_batch的影响

|  |  |  |
| --- | --- | --- |
| 模型 | 输入预训练词向量 | 不输入预训练词向量 |
| LSTM | 612 | 1032 |
| LDA-LSTM | 903 | 1408 |

### 4.5.4 LSTM、LDA-LSTM与LightGBM方法实验对比

研究问题（4）：不同模型方法推荐效果的对比

实验所用法条标签全集中有263条不同法条，即引用频次不低于50的法条，实验所用评估标准均为法条覆盖率CoverLaw值。

在多标签多分类方法下，对LSTM、LDA-LSTM与LightGBM进行实验，其实验结果如图4.10所示。随着K的增大，三个模型的CoverLaw值都持续增长。但LDA-LSTM在三种模型中表现最佳，LSTM模型其次，LightGBM的效果最差。不管K为何值LDA-LSTM都比其他模型的CoverLaw值高3%~4%，在K=18时，LDA-LSTM的CoverLaw值甚至达到了0.91。分析该现象背后的原因，本文认为由于加入LDA先验信息，并通过Attention机制额外关注与主题意思相近的词进而来进行法条推荐，此过程与人类注意力极为相似。自然场景下，普通人在阅读一篇裁判文书时，会将焦点放在与文书主题相关的词语上，从而对案件有更精确的了解。

图4. 10 多标签多分类下不同模型运行结果对比图

其次，LDA-LSTM进行多标签多分类所基于的向量为计算attention后的向量，有更多的注意力信息，而LSTM进行多标签多分类基于的向量为最后一个时间步的隐层输出向量。根据2.5节介绍，在输入序列过长的情况下，LSTM对序列的记忆会出现链状衰减，所以会导致LSTM的CoverLaw值低于LDA-LSTM。

对于LSTM与LDA-LSTM模型而言，都是采用端到端的深度学习网络，不用做任何特征工程，网络会自动学习“友好”的文本特征。而对于LightGBM而言，需要将文本通过特征工程转为向量，再通过LightGBM进行决策，特征工程设计的好坏将直接决定LightGBM模型预测的好坏。就本实验结果可以看出，LightGBM的预测效果最差，反映了以tfidf向量作为文本特征向量并不适合。

除了多标签多分类方法，本文还从多分类方法研究了模型的预测效果。当一篇裁判文书引用多个法条，如[法条1，法条2，法条3]，将该条记录拆分为三条记录。每条记录的案件基本情况都相同，但引用法条标记不同，最后得到每条记录只有一个法条标记的语料库。通过对该语料库进行单标签多分类，得到的实验结果如图4.11所示。LDA-LSTM模型的CoverLaw值最高，LSTM模型的值其次，LightGBM模型的值最小，其原因大致与多标签多分类情况下类似。

图4. 11多分类下不同模型运行结果对比图

### 4.5.5 多分类、多标签多分类方法的对比

研究问题（5）：多分类、多标签多分类方法对法条推荐效果的影响

实验所用法条标签全集中有263条不同法条，即引用频次不低于50的法条，实验所用评估标准均为法条覆盖率CoverLaw值。

实验结果如表4.11所示，当K=6与K=12时，多标签多分类方法的CoverLaw值普遍高于多分类方法CoverLaw值。对于LSTM、LDA-LSTM与LightGBM而言，当K=6时，多标签多分类方法的CoverLaw值比多分类方法高2%~3%；当K=12时，多标签多分类方法的CoverLaw值比多分类方法高2%~3%，LightGBM两者齐平；当K=18时，多标签多分类方法的CoverLaw值比多分类方法高1%~5%。本文认为，对于法条推荐的真实场景下每条法条而言，是否被推荐为一个二分类的问题。在一篇裁判文书中，如果裁判文书引用了多条法条，就需要检测多个目标，也就是多目标检测问题，对应着多标签多分类问题。

表4. 11 多分类与多标签多分类方法对比

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 模型 | K=6 | | K=12 | | K=18 | |
| 多分类 | 多标签  多分类 | 多分类 | 多标签  多分类 | 多分类 | 多标签  多分类 |
| LSTM | 0.71 | 0.73 | 0.82 | 0.84 | 0.87 | 0.87 |
| LDA-LSTM | 0.75 | 0.78 | 0.85 | 0.88 | 0.86 | 0.91 |
| LightGBM | 0.71 | 0.73 | 0.81 | 0.81 | 0.85 | 0.87 |

## 4.6 方法的优缺点分析

对于本文提出的两种法条推荐方法，都是基于深度学习的法条推荐方法。相比与传统机器学习算法如SVM、DecisonTree以及LightGBM而言，本文提出的法条推荐方法不需要做特征工程，而在网络训练的时候可以自发学习特征，由此避免了人工特征工程设计繁琐或设计不当的问题。而传统机器学习算法预测效果严重依赖于特征工程，如果特征工程环节出现问题，传统机器学习算法的预测效果可能会非常差。接下来本节将对LSTM以及LDA-LSTM模型的优缺点进行详细的分析。

基于LDA-LSTM的法条推荐方法优点：该方法相对单纯的LSTM而言，可以记忆更长的序列信息。以LDA主题向量作为先验知识，模仿人类注意力机制，重点关注裁判文书中与主题相关的词语。由于计算注意力向量时，提升了与主题相关词语的权重，而减少了与主题不相关词语的权重，使得在法条推荐更有针对性。其次，由于LDA-LSTM采用了Attention机制，对输入序列的每一个时间步都有关注，能够处理较长序列依赖的问题。而LSTM存在记忆链状衰减，对序列长度是有限制的。

基于LDA-LSTM的法条推荐方法缺点：模型较为复杂，需要更多的训练数据，训练复杂度较高，对GPU等计算资源要求较高。

基于LSTM的法条推荐方法优点：由于LSTM本质是循环神经网络，能较好的对序列数据进行语义分析。而相对于LDA-LSTM而言，模型复杂度低，能快速的搭建并学习，可以对模型参数进行细致调节。

基于LSTM的法条推荐方法缺点：虽然LSTM能处理序列信息，但序列太长时，由于记忆链状衰减的原因，仍然无法记忆很久之前的信息。

除此之外，经本文的实验证明，对于法条推荐问题，选择多标签多分类方法明显好于多分类方法。因为对于一篇裁判文书而言，存在多个法条引用的场景下，属于多目标检测问题，因此使用多标签多分类方法来解决法条推荐问题是一个很好的选择。

## 4.7 本章小结

本章主要介绍两种法条推荐方法的实验过程与实验评估，首先对实验采用的数据集进行介绍，然后介绍了多项法条推荐模型的评估标准，接着介绍了预训练LDA主题模型以及Word2vec词向量模型的方法以及法条推荐模型的超参数设置。在对比实验环节中分别展示了标准化标签集中法条总数对推荐效果的影响；法条推荐个数K对推荐效果的影响；预训练词向量对LSTM与LDA-LSTM的影响；LSTM、LDA-LSTM与LightGBM实验对比；多分类、多标签多分类方法的对比等一系列实验结果。最后本章分析了LSTM模型与LDA-LSTM模型的优缺点。

# 第五章 总结和展望

## 5.1 总结

本文以深度学习方法为基础，以XML格式的裁判文书为数据源，着重研究了基于深度学习的法条推荐方法。本文首先根据裁判文书本身的数据特点，提出了裁判文书特有的预处理方法，包括法律法条标准化以及法律专用停用库的构建，避免了通过大量人力阅读来标准化法律裁判文书以及构建法律专有停用词典。

然后本文详细介绍了两种法条推荐方法，分别是基于LSTM的法条推荐方法以及基于LDA-LSTM的法条推荐方法。第一种方法采用变长的LSTM网络，对输入的“查明事实段”序列内容进行语义分析，基于最后输出的隐层向量进行多标签多分类，以完成法条推荐任务。第二种方法引入“查明事实段”的LDA主题向量作为先验信息，通过Attention机制额外关注与主题相关的词，计算出注意力向量，并基于注意力向量进行多标签多分类。本文详细介绍了两种方法的模型框架以及训练方法。

在实验分析阶段，本文从多个角度对法条推荐方法进行评估，包括宏F值，法条覆盖率等。并选取六大案由的文书作为训练数据，对超参数设置进行了详细分析。同时通过实验对比了基于LSTM的法条推荐方法、基于LDA-LSTM的法条推荐方法以及基于LightGBM的法条推荐方法的推荐效果，进一步总结了各个模型优缺点。

## 5.2 展望

对于本文所提出的法条推荐方法仍有一些不足，还有提升的空间，具体如下：

（1）先验信息的引入问题：本文以LDA主题向量为先验信息，通过Attention机制将先验信息引入到深度网络中，模型结构较复杂，这对GPU以及内存等资源要求较高。因此，能找到另一种方法引入先验主题向量也是未来的工作重点。

（2）使用多种信息联合学习来完成法条推荐：案由本身也体现了案件基本情况的主题，可以案由预测以法条推荐最为两个任务进行联合学习，两个任务互相监督强化以提升法条推荐效果，是未来的一个工作方向。

（3）引入法条内容信息：本文进行法条推荐时，完全忽略了法条内容本身的信息。通过对法条内容本身的语义分析以及“查明事实段”的语义分析，可以进一步改善法条推荐的效果。

# 参 考 文 献

[Bansal T et al., 2016] Bansal T, Belanger D, McCallum A. Ask the gru: Multi-task learning for deep text recommendations[C]//Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016: 107-114.

[Cardellino C et al., 2017] Cardellino C, Teruel M, Alemany L A, et al. A low-cost, high-coverage legal named entity recognizer, classifier and linker[C]//Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law. ACM, 2017: 9-18.

[Feng Y et al., 2018] Feng Y, Ge J, Li C, et al. Statutes Recommendation Using Classification and Co-occurrence Between Statutes[C]//Pacific Rim International Conference on Artificial Intelligence. Springer, Cham, 2018: 326-334.

[García-Constantino M et al., 2017] García-Constantino M, Atkinson K, Bollegala D, et al. CLIEL: context-based information extraction from commercial law documents[C]//Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law. ACM, 2017: 79-87.

[Graves A et al., 2013] Graves A, Jaitly N, Mohamed A. Hybrid speech recognition with deep bidirectional LSTM[C]//2013 IEEE workshop on automatic speech recognition and understanding. IEEE, 2013: 273-278.

[Hyung Z et al., 2014] Hyung Z, Lee K, Lee K. Music recommendation using text analysis on song requests to radio stations[J]. Expert Systems with Applications, 2014, 41(5): 2608-2618.

[Luong M T et al., 2015] Luong M T , Pham H , Manning C D . Effective Approaches to Attention-based Neural Machine Translation[J]. Computer Science, 2015.

[Ke G et al., 2017] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree[C]//Advances in Neural Information Processing Systems. 2017: 3146-3154.

[Le Q et al., 2014] Le Q, Mikolov T. Distributed representations of sentences and documents[C]//International conference on machine learning. 2014: 1188-1196.

[M Lei et al., 2017] Lei M, Ge J, Li Z, et al. Automatically Classify Chinese Judgment Documents Utilizing Machine Learning Algorithms, *International Conference on Database Systems for Advanced Applications*. Springer, Cham, 2017:3-17.

[Michal Rosen-Zvi et al., 2004] RosenZvi, Michal, Griffiths, et al. The author-topic model for authors and documents. *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, 2012:487-494.

[Suchal J et al., 2010] Suchal J, Návrat P. Full text search engine as scalable k-nearest neighbor recommendation system[C]//IFIP International Conference on Artificial Intelligence in Theory and Practice. Springer, Berlin, Heidelberg, 2010: 165-173.

[Sundermeyer M et al., 2012] Sundermeyer M, Schlüter R, Ney H. LSTM neural networks for language modeling[C]//Thirteenth annual conference of the international speech communication association. 2012.

[Vaswani A et al., 2017] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems. 2017: 5998-6008.

[Wang C et al., 2011] Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles[C]//Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011: 448-456.

[Xu K et al., 2015] Xu K, Ba J, Kiros R, et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention[J]. Computer Science, 2015:2048-2057.

**[郭叶等，2017**] 郭叶,孙妹.指导性案例应用大数据分析——最高人民法院指导性案例司法应用年度报告(2016)[J].中国应用法学,2017(04):40-62.

[简璐瑶，2012] 简璐瑶. 基于领域本体的公安案情文本挖掘研究[D].南昌大学,2012.

[李凡等，2001] 李凡,鲁明羽,陆玉昌.关于文本特征抽取新方法的研究[J].清华大学学报(自然科学版),2001(07):98-101.

**[康东，2014] 康东. 中文文本挖掘基本理论与应用[D].苏州大学,2014.**

[王竹，2017] **王竹.《民法总则(草案)》若干法律规范去留问题大数据分析——以《民法通则》相应条文的司法适用大数据报告为基础[J].四川大学学报(哲学社会科学版),2017(01):149-160.**

[王继成等，2000] 王继成,潘金贵,张福炎.Web文本挖掘技术研究[J].计算机研究与发展,2000(05):513-520.

[王海亮，2017] 王海亮. 基于文本挖掘的法律咨询系统研究和实现[D].北京化工大学,2017.

[张保生，2001] 张保生.人工智能法律系统的法理学思考[J].法学评论,2001(05):11-21.

# 致 谢

光阴似箭，日月如梭，在南京大学软件学院两年研究生生活即将结束。在论文即将完成之际，我想对两年来给予我关心、照顾、鼓励和支持的老师，同学以及家人表示衷心的感谢。

首先我要感谢我的研究生导师葛季栋老师。葛老师博学多识，治学严谨，认真负责。非常感谢葛老师在过去两年中，对我悉心的教育与指导，耐心地指出我工作中的不足之处，并教导我如何解决问题，让我从您那里学到很多处理问题地方法。

同时，我要感谢在学习生活中给予我极大帮助的李传艺老师，李老师对科研的热情以及敢于克服困难的科研精神令我印象深刻，受益匪浅。感谢已经毕业的雷妙妙学姐，对我毕业设计的关心和指导，给了我很大的帮助。同时感谢冯奕同学，赵斯蒙同学，高永伟同学以及朱鹏同学，在我沮丧失意的时候能给我很大的支持与鼓励，你们刻苦钻研，勤奋踏实，乐观积极的品质都值得学习。

感谢和我朝夕相处的可爱的室友们以及同学们，有你们的陪伴让我一路走来都不感觉孤单。最后，我要感谢一直以来默默关心我和支持我的家人，希望你们永远幸福健康。

# 附 录

非必须成分。列出与本文内容相关的附录文档，一般为篇幅较长的技术规范或者数据表。

# 参 与 项 目

非必须成分。列出在攻读硕士论文期间参与的项目。

# 版权及论文原创性说明

任何收存和保管本论文的单位和个人，未经作者本人授权，不得将本论文转借他人并复印、抄录、拍照或以任何方式传播，否则，引起有碍作者著作权益的问题，将可能承担法律责任。

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，本论文不包含其他个人或集体已经发表或撰写的作品成果。本文所引用的重要文献，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。

作者签名：

日期： 年 月 日