电子科技大学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

硕士学位论文

MASTER THESIS

论文题目 网络舆情信息识别与分析的关键技术研究

学科专业 计算机科学与技术

学　　号

作者姓名

指导教师

分类号 _____ 密级 _____

UDC [注1] _____

学　位　论　文

网络舆情信息识别与分析的关键技术研究

（题名和副题名）

***

（作者姓名）

指导教师 _____

电子科技大学　　成都

_____

_____

（姓名、职称、单位名称）

申请学位级别 硕士　学科专业 计算机科学与技术

提交论文日期 _____ 论文答辩日期 _____

学位授予单位和日期 电子科技大学

答辩委员会主席 _____

评阅人 _____

注 1：注明《国际十进分类法 UDC》的类号。

Research on Key Technologies of Network Public Opinion Information Identification and Analysis

A Master Thesis Submitted to

University of Electronic Science and Technology of China

Discipline: **Computer Science and Technology**

Author: ***

Supervisor: ***

School: **School of Computer Science & Engineering**

独 创 性 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

签名：_____ 日期：　　年　　月　日

关于论文使用授权的说明

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

签名：_____　　导师签名：_____

日期：　　年　　月

摘要

随着我国互联网技术的快速发展，人们逐渐使用电子设备通过网络通道来进行日常的工作和交流，广大网民成为了网络舆情信息传播的主要介质，网络中的舆情信息爆发式增多。冗长的舆情数据不仅严重浪费舆情信息分析人员的时间和精力，而且其内容中的不良言论也会给社会稳定带来影响。此外，在海量鱼龙混杂的网络数据中存在大量对相关部门有价值的舆情信息，如何获取并高效的分析这些数据从而帮助有关部门更好的了解社情民意是一个亟待解决的问题。

基于以上问题，本文对网络舆情信息识别与分析中所涉及到的文本摘要技术和文本分类技术进行了相关研究与探索。本文的主要工作包括：

1、针对舆情信息文本过长以及信息中存在主观情感内容的问题，基于带注意力机制的 Seq2Seq 模型实现了文本摘要模型，为每条舆情信息生成一个简短的摘要信息，将舆情信息简洁化，并使用 Coverage 机制解决模型生成过多重复词语的问题。

2、针对目前生成式文本摘要模型对于文本主题信息利用较少的问题，使用一种有监督算法提取出文本的关键词信息，并利用此关键词信息对注意力机制进行改进，使模型对文本主题信息更加敏感，从而使得模型的效果得到提升。

3、针对目前大多数生成式文本摘要模型都是从词级别或者字符级别对原文进行编码的问题，提出了一种双编码器文本摘要模型，同时从词级别和子句级别对原文进行编码，使得解码器所使用的语义向量包含的信息更加丰富，模型生成的摘要更加准确。

4、针对文本摘要模型编码器输入序列过长，RNN 编码器在编码的过程中逐渐丢失原文中靠前位置编码信息的问题，提出了一种双阶段式文本摘要模型。首先利用有监督算法将原文中与文本主题更为相似且含有更多文本关键词的子句挑选出来，然后将其作为生成式摘要模型的输入进行第二阶段的训练，在减少网络时空开销的同时提升了模型效果。

5、针对舆情信息过于零散和杂乱，难以从中获得高层次有价值信息的问题，使用文本分类技术将信息结构化。并以成都地区网络警情信息为例进行分析，根据其数据特点基于 CNN 设计并实现了一种新颖的文本分类模型 SPCNN，其分类效果优于其他对比模型。

关键词：网络舆情，文本摘要，双编码器，双阶段，文本分类

ABSTRACT

With the rapid development of Internet technology in China, people are gradually using electronic devices to do daily work and to communicate through network channels. The netizens have become the main media for the spread of public opinion information on the Internet, and public opinion information on the Internet has exploded. The lengthy public opinion data not only seriously wastes the time and energy of the analysts of public opinion information, but also brings impact in the negative content to social stability. In addition, there are a lot of public opinion information which is valuable to relevant departments in the mass mixed network data, so how to obtain and efficiently analyze these data to help them better understand social conditions and public opinion is an urgent problem to be solved.

Based on the above problems, related research & exploration on text summary technology and text classification technology involved in the identification and analysis of Internet public opinion information have been conducted in this paper. The main research contents of this paper are as follows:

1) Aiming at the problem that the public opinion information is too long and contains subjective emotional content, a text summary model is implemented based on the Seq2Seq model with attention mechanism. With this model, a brief summary information is generated for each piece of public opinion information, and the information is simplified. In addition, the Coverage mechanism is used to solve the problem of generating too many repeated words of the model;

2) Aiming at the problem that the current abstractive summary models make less use of text subject information, a supervised algorithm is used to extract the keyword information of the text, and this information is used to improve the attention mechanism in the model so that the model is more sensitive to the text subject information and the effect of the model is improved;

3) Aiming at the problem that most abstractive summary models encode text at the word level or character level, a double-encoder text summary model is proposed. The text is encoded at the word level and clause level in this model, which makes the context vector uesd by the decoder contain more information and the generated summary is more accurate;

4)Aiming at the problem that the input sequence of the text summary model

encoder is too long, and the RNN encoder gradually loses the previous position encoding information in the text during the encoding process, a two-stage text summary model is proposed in this paper. Firstly, a supervised algorithm is used to select clauses in the original text that are more similar to the text topic and more text keywords are contained, and then input these clauses into the abstractive summary model for second-stage training. The network space and time overhead is reduced while the model performance is improved;

5)Aiming at the problem that public opinion information is too fragmented and cluttered, and it is difficult to obtain high-level valuable information, the information is structured using text classification technology in this paper. Taking the police information of Chengdu Public Security Bureau as an example, according to its data characteristics, a novel text classification model SPCNN was designed and implemented based on CNN, and its classification effect is better than other comparison model's.

Keywords: Internet public opinion, Text summary, Double-encoder, Two-stage, Text classification

目录

第一章 绪论

1.1 研究背景以及意义

随着我国社会经济以及互联网技术的高速发展，人们逐渐使用电子设备通过网络通道进行日常的交流、工作和信息获取等。2019 年 8 月 30 日，中国互联网络信息中心（CNNIC）在北京发布第 44 次《中国互联网络发展状况统计报告》。《报告》指出，到 2019 年 6 月，我国网民的数量比 2018 年年底增加了 2598 万，网民总数量达到了 8.54 亿，其中手机网民的数量达到了 8.47 亿，且网民通过手机上网的比例高达 99.18%。各种网络平台及社交媒体的快速发展使得任何人在任何地方都能够平等、自由的发表自己对于任何事物的看法，广大网民成为了网络舆情信息传播的主要媒介。

目前网络上的许多舆情信息都是先由一些商业网站，如网易新闻、新浪新闻等以及微博上的一些大 V 进行撰写或者收集，然后报道给其用户或者粉丝。由于网络的自由性，这些信息在很大程度上都包含一定的主观色彩甚至是不良言论，相关工作人员如若不能对网络上的这些信息进行及时、有效的处理，那么网民的不良情绪就会被逐渐地诱发，甚至会引起一部分民众的过激反应和违法行为，导致社会稳定受到威胁。此外，这些信息会被其他网站以及网民不断地转发和再编辑，则网络中会存在大量关于同一事件的不同舆情信息，由于社会生活节奏的加快，网民们难以有充足的时间和精力对海量的信息以及过长的舆情报道进行筛选、阅读和思考。

网络舆情信息虽然给社会的安稳及人们的生活带来了一定困扰，但相关工作人员若能对其中有价值的信息进行合理的挖掘和利用，则他们的工作会获得很大的帮助。如通过对网上大量舆情信息的获取、整理和分析，相关工作人员可以从一个更高的层面对社情民意有一个了解，则他们的工作方向会更加明确，工作内容会更加具有针对性。

对于网络舆情信息的这些特点，社会管理人员意识到必须利用相关技术对舆情信息予以处理，但由于网上的信息量十分巨大，仅凭借传统的人工方式去收集、整理以及处理这些海量信息简直天方夜谭。所以，必须加强对舆情信息处理技术的研究，形成一些自动化、智能化的舆情信息处理方法，对网络中的舆情事件进行及时地应对，从而由原来的被动防堵处理方式转化为主动的梳理和引导[1]。

文本摘要技术旨在通过对原文内容的抽取或者理解后再生成，得到一个相对于原文而言篇幅更精简、主题更明确的文本内容。使用文本摘要技术为每条舆情信息生成一个简短的摘要，用户或者舆情信息处理人员通过阅读这些简短却能充

分代表文本主题的内容就可以对当下的热门事件有一个了解，使得获取信息的效率得到了提高。此外，生成式文本摘要技术得到的文本一般只会陈述事件事实，而不会带有情感色彩内容，这就使得网民们可以以一个更加客观的态度去看待事件，避免信息中不良评论以及不健康情绪的持续扩散。

文本分类技术旨在通过对文本进行分析，将文本按照提前设定好的类别进行归类[2]，这可以帮助相关舆情信息分析人员基于一个结构化的框架去挖掘舆情信息中有价值的内容。如相关部门可以从“城市形象、执法服务、党政干部、工商质监、教育相关”等不同的方面对舆情数据进行统计和分析，从而对当前社会中所存在的民生、民权、民主以及社会治安等问题有一个更加清晰的认识，然后对症下药，更好的维护社会安稳。

通过以上分析可知，运用自然语言处理中的文本摘要和文本分类等技术对网络中海量的舆情信息进行识别、处理、分析和利用是非常具有现实意义的。

1.2 文本摘要研究现状

1.2.1 文本摘要概述

文本摘要技术从上世纪 50 年代以来，已经发展了将近 70 年。在这 70 年中，中外学者从不同角度对文本摘要任务进行了探索，使得目前文本摘要技术虽然没有像其他自然语言处理技术一样成熟，但也得到了快速的发展。文本摘要技术从不同角度可以有不同的分类，具体如表 1-1 所示。本文所研究的生成式文本摘要相对于抽取式摘要而言，它所生成的摘要内容更加的简短，概括性也更高。

表 1-1 文本摘要技术分类

分类方式	类别	特点
文档数量	单文档文本摘要	根据一篇文档中的内容进行摘要的生成。
	多文档文本摘要	根据多篇描述同一事件的文档中的内容进行摘要的生成。
实现方式	抽取式文本摘要	将文档中的重要内容抽取出来重新组合进行摘要的生成。
	生成式文本摘要	对文档中的内容进行语义层面的理解之后重新生成新的语句进行摘要的生成。
使用技术	传统文本摘要	使用统计学、图模型、代数论等方法进行建模解决文本摘要问题。
	非传统文本摘要	使用机器学习、深度学习等方法进行建模解决文本摘要问题。

文本摘要技术在众多中外学者的不断研究下有了很大的提升，但是目前使用机器自动生成的摘要与人工提取出的摘要的质量相比，还有很大的进步空间。造成这种差距的主要原因有以下几点：

a) 对于机器学习模型以及深度学习模型而言，数据集的数量和质量对其效果有很大的影响，而目前关于文本摘要任务方面的数据集却较少，尤其是在中文数据集方面。数据的不充分使得模型的泛化能力得不到提升，而且，究竟需要多大的数据集才能满足模型训练的需求，目前也没有一个准确的理论依据。
b) 自然语言处理任务中对文本进行理解时，首先需要根据所使用的数据集中的文本建立词汇表或者字汇表，然后将文本根据字符或者词语进行向量化表示，最后再根据这些编码进行一系列运算得到文本的语义向量表示。在这个过程中，OOV（Out Of Vocabulary,未登录词）问题以及向量表示的准确性问题一直没有得到彻底地解决。
c) 文本摘要任务的结果是一段文本，由于大量近义词的存在以及语言的多样性，所以一条文本的正确摘要并不是唯一的，但目前文本摘要任务评测方法都是基于序列匹配规则对结果进行评估的，所以其灵活性有待提升。

1.2.2 国外研究现状

国外最早关于文本摘要技术的研究是从 Luhn 等人[3]提出的一种抽取式摘要模型开始的，他们对文本统计词频，然后将某个词频区间中的词语视作该文本的重要词语集合并根据子句中出现重要词语的数量对每个子句进行打分，最后根据子句的得分选取出若干个子句进行重新组合作为原文的摘要内容。之后，Baxendale 等人[4]通过研究文本主题与子句位置的关系发现，在与文本主题最相关的子句中，子句位置有 85%的概率位于段首位置，有 7%的概率位于段尾位置。为此，他们将子句位置这一因素加入了子句打分机制中，提高了摘要内容与文本主题的相关性。Salon 等人[5]通过探索设计出了一种新的确定词语权重的方法，TF-IDF。这种方法同时考虑词语在一个充分大的语料库中以及原文中出现的频率来确定它对该文本的重要性，然后基于此关键词信息挑选内容，提高了抽取内容的准确性。之后，受有监督机器学习方法的影响，Kupiec 等人[6]将抽取式文本摘要任务视为一个二分类问题，使用 Bayes 分类器将重要子句抽取出来，摒弃了人工为子句的不同特征设置不同权重的方法。

2006 年，Hinton 等人提出了深度置信网络[7]，深度学习被学者们不断地探索、应用在不同图像任务及自然语言处理任务上。2012 年，Liu 等人[8]使用深度学习设计了一种多文档摘要模型，获得了比当时大多数模型更为出色的表现效果。Rush 等人[9]基于 Encoder-Decoder 框架[10]构建了一个生成式文本摘要模型。其中，Encoder 是根据 Bahdanau 等人[11]在机器翻译中提出的基于注意力的编码器建模

的，Decoder 使用了 Bengio 提出的神经语言模型[12]来实现。他们的模型在 DUC-2004 数据集以及 Gigaword 数据集上获得了比其他对比模型都高的 ROUGE[13]得分。之后，Facebook 研究机构的 Chopra 等人[14]在 Rush 等人的研究基础上进行了改进，用卷积神经网络（Convolutional Neural Network,CNN）实现 Encoder，并且使用循环神经网络（Recuurent Neural Network,RNN）来实现 Decoder，进一步提升了模型的性能。2016 年，IBM 的 Nallapati 等人[15]在其研究中提出了许多建模时的技巧：在 Encoder 的嵌入层使用文本的多种特征以获得更丰富的原文信息；编码时，使用层次注意力捕获文本结构信息；解码时，在生成词语的同时使用指针机制从原文中复制一个词语然后决定预测词语的分布以解决低频词以及未登录词问题。加入了这些技巧后，模型在 Gigaword 数据集和 Duc-2004 数据集上都取得了当时最好的效果。此外，他们还针对当前英文文本摘要数据集中的标准摘要都只有一句话的问题，基于 Hermann 等人[16]贡献的问答任务数据集提出了 CNN-DM 文本摘要数据集。See 等人[17]针对 OOV 问题在使用了注意力机制的 Seq2Seq 模型中加入了指针生成网络，不同于 Nallapati 等人只从原文中取出概率最大的词语进行“复制”的做法，他们根据每个解码时刻的注意力分布决定对原文中每个词语的复制概率。此外，他们还针对使用注意力机制的模型会生成重复词语的问题提出了 Coverage 机制，并在模型的损失函数中添加了覆盖损失以在解码的每一步惩罚那些在之前解码时刻已经贡献了较多注意力的词语。此模型相比于 Nallapati 等人的模型，在 CNN-DM 数据集上的得分提高了 3%左右。随后 Paulus 等人[18]将强化学习（Reinforced Learning,RL）与深度学习相结合，提出了基于深度强化模型的生成式摘要方法，为后人在强化学习上的探索奠定了基础。近期，由于谷歌智能研究院发布的预训练模型 BERT[19]在语言理解上表现出的强大能力，一些学者开始基于 BERT 进行了探索，如斯坦福大学的 Khandelwal 等人[20]以及 Subramanian 等人[21]。

1.2.3 国内研究现状

相对于国外而言，国内关于文本摘要技术的研究起步较晚，发展也较为缓慢。最早在此领域进行探索和研究的是苏海菊等人[22]，他们使用文本摘要技术实现了对中文科技文献的文摘进行自动编写。之后由于当时中文文本摘要公开数据集的缺失，国内的研究进程相对于国外而言较为缓慢。2015 年，程园等人[23]考虑文本中的句子位置、子句相似度、词频、提示性短语等因素，使用回归模型进行建模，并且使用人民网上 5 个新闻领域的 800 篇新闻对模型进行了训练，开启了国内学者在中文文本摘要上的研究。郭艳卿等人[24]对多文档摘要进行了研究，由于他们使用的数据是从新浪、搜狐、网易、人民网等获得的 60000 篇无重复报道，而新闻报道一般都基于时间线进行编辑的，所以他们按照时间节点对文本中的内

容进行划分，然后根据自行设计的全局权重机制和局部权重机制对其进行评估，最终抽取出得分最高的内容。

2016 年 6 月，Hu 等人提出了第一个中文文本摘要公开数据集 LCSTS（A Large Scale Chinese Text Summarization Dataset）[25]，他们在两种 Seq2Seq 模型上分别使用字符级别嵌入及词级别嵌入进行了实验，并使用 Lin 等人提出的 ROUGE 评估标准进行了评估，为未来在中文文本摘要方面的研究工作提供了一个基准。Gu 等人[26]为了解决 OOV 问题，提出了 COPYNET，使得模型在解码的每一步既可以选择生成模型词汇表中的词语又可以选择从原文中复制词语，在 LCSTS 数据集上展现了很好的性能。之后，为了解决最大似然这一训练目标只能在词级别对模型进行约束，以及现有的模型严重依赖于数据分布的问题，Ayana 等人[27]提出了最低风险训练策略，这一策略直接根据模型的评估值在句子级别对模型的参数进行优化。实验结果表明，他们的模型在两个英文数据集 Gigaword、DUC-2004 以及中文数据集 LCSTS 上都取得了最好的效果。周才东等人[28]认为编码器编码得到的信息不充分是因为现有模型对文本的高层次特征缺乏利用，因此他们先使用局部注意力结合卷积神经网络实现了一个特征提取器，然后将特征提取器的输出作为模型编码器的输入，最后在解码的每一步使用全局注意力机制进行生成词的预测。此模型在使用词级别进行嵌入时，在 LCSTS 数据集上得到了很好的效果。第二个中文文本摘要公开数据集 NLPCC2017task#3[29]是 CCF 在 2017 年举办 NLPCC 全国性评测比赛时提出的，相对于 LCSTS 而言，此数据集较小，只包含了 5 万个<原文，标准摘要>序列对和 3000 条原文本数据。当时在此数据集上取得效果最好的是 Hou 等人[30]，他们参考 Sennrich 等人[31]在机器翻译中的做法，在模型中使用了 BPE 编码（Byte Pair Encoding,双字节编码）[32]和 Subword-embedding 以降低模型词汇表的大小，并且在解码的每一步同时使用全局注意力以及解码注意力以便生成的摘要内容准确无重复。之后，他们又提出了一种主题信息融合模型[33]，通过 TextRank 方法提取出文本的关键词，然后将原文信息和关键词信息同时编码到文本的语义向量中以获得更加丰富准确的原文信息，此模型在 NLPCC2017task#3 数据集上获得了目前最好的效果。

1.3 文本分类研究现状

1.3.1 文本分类概述

文本分类技术就是将一个文本库中的文本按照提前设定好的类别归整到其中一个类别或者多个类别中，从而将数据结构化。文本分类技术从是否有监督、使用的技术手段、使用的特征等不同角度看，可以有不同的分类方式，具体如表 1-1 所示。

表 1-1 文本分类技术分类

分类方式	类别	说明
是否有监督	有监督学习算法、无监督学习算法、半监督学习算法	有监督算法如 SVM、Bayes、CNN、RNN 等；无监督算法如 K-Means 算法；半监督算法所使用的数据集中只有少量数据有标签，大部分数据是无标签的。
使用的技术	传统机器学习算法、深度学习算法、迁移学习算法	传统机器学习算法基于人工设计的知识规则进行学习，如 SVM、Bayes 等；深度学习算法可以使用网络的自学习能力提取到数据的特征，如 CNN、RNN 网络等；迁移学习主要利用源领域知识辅助目标领域任务的解决，如 BERT、GPT[34]。
使用的特征	基于文本标题进行分类、基于文本内容进行分类、添加外部知识特征进行分类	添加的外部知识主要有词语的词性信息、词语的词频信息、文本的关键词信息、文本的主题信息等。

20 世纪 60 年代，文本分类技术被提出，其初期的研究工作主要使用基于知识工程的传统方法。到了 90 年代后，机器学习(Machine Learning,ML)方法被逐渐的探索和使用，此时期的分类问题被拆分为特征工程和分类器的使用两部分。2006 年，深度学习（Deep Learning,DL）因为置信网络在 Hinton 等人的研究中表现出的优越成绩而被重视，它在图像研究和语音研究方面所表现出的优秀的特征提取能力被自然语言处理领域的学者所关注，于是深度学习被逐渐地用于解决各类文本分类问题。近期，基于 GPT 和 BERT 等大型预训练模型的提出，自然语言处理任务中所使用的字向量或者词向量的质量得到了提升，则不同任务所使用的模型提取出的数据特征更加具有表征能力，文本分类技术得到了很大的发展。虽然目前国内外对于文本分类技术的研究已经较成熟，但仍有一些挑战尚未完全完成，具体如下：

a) 机器学习，尤其是深度学习，是基于数据驱动的，而目前模型训练所使用的数据集都是人工进行标注的。一方面，人工标注的数据具有一定的主观性，则数据的真实标签有待商榷；另一方面，人工标注数据的数据量也无法确定。因为标注的数据较多会耗费巨大的时间和精力，而标注的数据较少会导致采样的数据无法代表数据的真实分布，从而导致模型的泛化能力较差，模型效果降低。
b) 对于中文文本分类任务而言，文本预处理中的分词问题是一大难题，因为它是文本表示的核心步骤。目前的分词工具 jieba、HANLP 等虽然可以识别人

名、地名等实体词，但是由于目前网络用语的快速更新，这些工具很难维持很好的分词效果，则文本的表示会受到一定影响，从而文本分类的准确率也会降低。

c) 目前神经网络的效果虽然很好，但它被当做黑盒来使用。研究人员对其工作原理了解有限，模型效果的好坏也没有夯实的理论依据支撑，且模型参数值的设置、激活函数和梯度调节算法的使用经常靠经验来进行选择。

1.3.2 国外研究现状

国外关于文本分类技术的研究可以追溯到上世纪 60 年代，此时学者们主要通过根据领域专家建立的规则及手工建立的文本分类器解决文本分类问题。90 年代后，机器学习的兴起使得学者们开始在文本特征选择以及分类器的使用方面进行探索。Dasgupta 等人[35]提出了一种无监督的特征选择算法，并通过对比试验证明了其方法的可行性。Gunnemann 等人[36]提出了一种数据降维的方法，其主要思想是对子空间进行聚类，从而降低分类器的计算开销。Vries[37]和 Liaw[38]等人分别从数据降维、加快训练速度方面对 KNN 分类算法进行了改进。Hasan 等人[39]使用 Apriori 算法提取出频繁项目集并将其作为决策树的训练规则进行输入，提升了决策树算法在文本分类任务上的效果。

2006 年，Hinton 提出了深度置信网络后，卷积神经网络、循环神经网络等相继被用在文本分类任务建模中。2014 年，Kim 等人[40]提出了 TextCNN 网络，使用基于卷积层、池化层等结构构建的卷积神经网络进行文本分类，为之后神经网络在文本分类领域的应用奠定了坚实的基础。由于 TextCNN 中的降采样技术使用的是最大池化（Max Pooling），它虽然将最具有表征力的特征保留了，但是也丢失了一部分数据信息，为此， Kalchbrenner 等人[41]提出了 DCNN（Dynamic Convolutional Neural Network）网络。Kalchbrenner 根据循环神经网络和递归神经网络的优缺点，设计了一种 k-max-pooling 方法以捕获文本句法结构信息，其中 k 代表在进行池化操作的时候将其中最大的 k 个值保留下来，且 k 会随着网络层数的加深而减小。Lai 等人[42]基于 CNN 和 RNN 各自的优缺点提出了 RCNN。模型的主要思想是先使用双向 RNN 进行编码，获得语义和语序信息，然后将其与各自时刻的词表示进行拼接再经过一个全连接层，以此作为网络的卷积层，之后的做法和 TextCNN 一致。Joulin 等人[43]出于节省时间的动机，提出了一种同时进行词向量训练和文本分类任务的工具 FastText，该网络结构与 Word2vec 中的 CBOW 十分相似。随着注意力机制在机器翻译中被广泛使用，Yang 等人[44]将其应用到了文本分类任务中，提出一种层次注意力文本分类网络 HAN。该网络首先使用双向 GRU 在词语级别上进行编码，然后使用注意力机制获得每个词语的权重，基于此权重和之前编码得到的词向量计算得到每个子句的句向量，随后再

次使用注意力机制，获得每个子句的权重，最后使用加权求和得到的向量作为文本的特征向量。Gao 等人[45]将 HAN 网络中的 RNN 替换成 CNN 进行了相关研究和测试。此外，Zhou 等人[46]将双向 LSTM（Long Short-Term Memory,长短期记忆网络）与注意力机制进行了结合。首先使用双向 LSTM 获得各个词语的语义向量，然后基于此获得各子句向量，再使用注意力机制获得各词语及各子句的注意力，并使用子句注意力对其所包含的词语的注意力进行调整，最终根据词级别的注意力模型获得文本的语义向量并使用 Softmax 回归进行分类。此模型在 NLPCC2013 跨语言情感分类数据集上获得了最好的效果。

1.3.3 国内研究现状

在传统算法研究与应用方面，李荣陆等人[48]从减少训练集样本数量的角度对 KNN 算法进行了改进。刘赫等人[49]提取出了多种文本特征，然后对它们赋予不同的权重，最后根据特征项及其权重进行计算得到文本表示并预测得到其类别信息。张杰慧等人[50]结合 SVM 对蚁群算法中的观察半径制定了动态调整策略，提高了算法效率，降低了空间和时间开销。李楠等人[51]应用决策树中的 ID3 算法对图书馆的数字信息进行了结构化整理。单丽莉等人[52]为了提取出更具有区分能力的文本特征，用调整交叉熵期望值的方法改进了特征提取算法，提高了模型的分类效果。蒋胜利等人[53]基于遗传算法提出了一种新颖的特征选择方法，在保证特征区分能力的同时在一定程度上解决了特征维度较高的问题。李静等人[54]提出了类别虚核的概念，即根据每类数据的特征项生成该类别的虚核，然后根据各个类别虚核对数据引力的强弱预测其所属类别，实验证明此算法效果优于朴素贝叶斯和 KNN 算法。

在深度学习的研究方面，赵明等人[55]基于改进的 Word2Vec 词向量构建文本向量并用 LSTM 作为分类网络对饮食文本进行了自动分类，帮助人们健康饮食。梁斌等人[56]针对 RNN 等序列模型训练时间开销大的缺点，提出了一种具有多注意力的 CNN 网络，在降低时间开销的同时得到数据的内容信息和结构信息，最终获得比其他对比模型都好的效果。卢玲等人[57]针对长文本分类中存在的噪声信息过多问题，提出了一种双阶段的模型。首先将一条数据按段落切分并获得向量表示，然后构建一个注意力模型以获得每个段落的注意力，随后根据段落注意力的均方差对段落进行过滤，最终将保留的内容及对应的注意力矩阵作为 CNN 网络的输入。实验结果证明使用过滤后的内容进行分类的效果更好。陈珂等人[58]在对微博内容的情感进行分析时对文本中的情感词语进行了特殊词性标记，然后将词语的词性信息以及位置信息作为特征加入到网络的输入通道中以使模型更加关注文本中的情感词语。模型在 COAE2014 数据集和 MBD 数据集上都获得了较好的性能。杜雨萌等人[59]通过提取出微博文本的主题信息并将其作为额外的

特征输入到 CNN 网络中，降低了文本中噪声信息对模型的影响，使模型能够更准确的识别用户兴趣。刘腾飞等人[60]使用双向 RNN 结合 CNN 并使用 Softmax 回归进行分类的方式构建模型，在六个分类任务上进行实验证明了其模型的有效性。王根生等人[61]通过改进的 TF-IDF 算法获得文本中每个词语的重要性并使用 Word2Vec 获得文本中词语的词向量，然后对这两者进行加权求和计算得到文本的向量表示，最后再基于此向量表示使用 CNN 网络获得更抽象的文本特征进行文本分类。陈志等人[62]针对文本分类中训练数据不平衡造成模型效果差的问题，根据每个类别的数据量为其设置了一个类别权重并将此权重加入到损失函数的计算中，改善了分类结果向多数类别倾斜的问题。段丹丹等人[63]借助预训练模型 BERT 获得文本的子句向量并使用 Softmax 回归模型进行训练，在测试集上的 F1 值达到了 93%，比使用 CNN 网络建模进行分类的效果高了 6 个百分点。

1.4 研究目标与主要内容

本文首先梳理了目前网络舆情信息的特点，阐述了舆情信息给广大网民和相关工作人员带来的方便和困扰以及给社会发展带来的两面性，然后分析了其造成不利情况的原因，即当前互联网技术的飞速发展，使得任何人在任何时候都可以平等自由的对网络上的事物发表评论，并且可以对他人发表的内容进行转发或者再编辑后发布，从而导致舆情信息数量暴增，舆情信息内容过于丰富，以及对于同一事件的舆情信息层出不穷，尤其是当时的热门事件，如“成华区摔狗事件”、“工作 996，生病 ICU”等，这对平台用户以及舆情信息分析人员都造成了严重的影响。为此，本文致力于使用自然语言处理领域中的相关技术对海量的舆情信息进行识别与分析，以帮助广大网民及舆情信息处理人员更高效的利用其中所蕴含的价值。

本文首先研究了单文本摘要技术，为舆情信息中的新闻报道形成摘要，使用户或者舆情信息处理人员通过阅读这些简短的、具有代表性的且不含有情感色彩的内容就可以对当下的热门事件有一个了解，在提高阅读效率的同时以一个更加客观的角度去看待事件。其次，本文研究了文本分类技术，将文本按照相关人员提前设定好的类别进行归类总结，从一个结构化的角度去挖掘舆情信息中有价值的内容，使相关工作人员从一个更高的层次去把握当下社会所存在的问题，进而对症下药，更好的维护网络社会的健康和现实社会的安稳。本文的主要研究内容有：

a) 分析了当前网络舆情信息的特点，针对其信息量过大、信息过于零散的问题，对自然语言处理中的相关技术以及相关实现工具进行了深入的调研、分析和学习；

b) 针对舆情信息中信息量过大、新闻报道的文本太长以及信息阅读者时间精力有限的问题，研究了文本摘要技术并提出了三种基于 Seq2Seq 的文本摘要模型。第一种是对模型中的注意力机制进行改进，使用文本的关键词信息对原有的注意力权重进行调整，使模型对文本的主题信息更加敏感，从而使模型生成的摘要更加准确；第二种是针对目前大多数模型都是从原文的词级别或者字符级别对原文进行编码的问题，提出了一种双编码器模型，同时从词级别和子句级别对原文进行编码，从而使得解码器在解码每一时刻使用的上下文向量的表征能力更强；第三种是针对部分文本过长，模型难以准确的捕捉到文本主题的问题，提出了一种双阶段式文本摘要模型，首先设计了一种抽取式摘要方法，将原文中与文本主题更为相关且包含更多文本关键词的内容挑选出来，然后将其作为生成式摘要模型的输入进行第二阶段的训练。

c) 针对舆情信息过于零散，难以从中获得高层次有价值信息的问题，本文以成都地区公安机关对警情信息处理的相关需求为例进行了相关研究。首先，使用爬虫技术从各个网络平台中获得了警情数据，随后对数据进行处理，然后根据此警情数据的特点，基于 CNN 设计并实现了一种文本分类模型。

d) 对于以上提出的模型，设计对比试验，在公开数据集上或自己准备的专业领域的数据集上，使用已有的评价方法对其效果进行评估和分析。

1.5 论文的组织结构

本文的内容总共有 6 个章节，每个章节的结构和内容如下：

第一章为本文的绪论部分，此部分首先对网络舆情信息的现状进行了介绍，紧接着阐述了文本摘要技术以及文本分类技术对于舆情信息处理的重要意义，然后分别对文本摘要技术及文本分类技术进行了概述并指明了目前所存在的问题，并且从国外和国内两个方面介绍了其各自的研究现状，最后对本文的研究目标以及主要的研究内容进行了总结。

第二章对本文研究内容中所涉及到的相关技术背景进行了介绍，包括基础的框架、常用的网络模型及相关的理论知识。首先介绍了深度学习中常用的网络模型卷积神经网络 CNN、循环神经网络 RNN、RNN 的变体 LSTM 及双向循环神经网络 BRNN，然后对对文本摘要任务常用的模型和技术手段进行了介绍，如基于 Encoder-Decoder 的 Seq2Seq 模型和集束搜索等。

第三章主要对本文所提出的三个文本摘要模型进行了详细的介绍。首先为了使模型对于原文中的主题信息更加敏感，本文基于文本关键词信息对注意力机制进行了改进。其次，针对使用词语级别的编码器获得的语义向量不准确的问题，提出了一种双编码器摘要模型，分别从词语级别和子句级别对原文进行编码。随

后，针对文本摘要数据原文中存在大量与摘要内容无关或者冗余的内容，提出了一种双阶段文本摘要模型，首先从原文中提取出关键内容，然后根据提取出的内容进行生成式文本摘要的学习。最后对本文提出模型的实现中涉及到的重要方法进行了介绍。

第四章为文本分类模型的介绍。此章针对成都地区舆情数据中的警情数据进行了分析，然后针对此数据集的特点提出了一种新颖的文本分类模型-子序列建议卷积神经网络模型（Subsequence Proposed Convolutional Neural Network, SPCNN），详细说明了模型的训练过程并对模型实现中的关键方法进行了介绍。

第五章为本文的模型测试部分。对于文本摘要和文本分类两个模块，分别介绍了其各自所使用的数据集以及效果评价方法，然后对本文所提出的模型进行了相关测试和分析，最后将本文模型与其他模型进行了效果对比。

第六章是本文的总结与展望部分。首先对本文中所完成的研究以及工作进行了总结，然后针对这些工作中目前仍需要进一步研究和改进的地方进行了分析和展望。

第二章 相关背景知识

本章主要对本文研究内容中所涉及到的相关技术背景进行了介绍，包括基础的网络结构和模型。首先介绍了深度学习中常用的卷积神经网络、循环神经网络、长短时记忆网络以及双向循环神经网络（Bidirectional Recurrent Neural Network,BRNN），然后对文本摘要任务中的基础模型 Seq2Seq 以及解码时所用的集束搜索算法进行了介绍。

2.1 机器学习模型

2.1.1 卷积神经网络

卷积神经网络在计算机视觉中的广泛应用使学者们积极探索其在自然语言处理中的应用方式。通过将每个词语表示为一个实数向量，便可以将一条自然语言表示的文本转换为一个数据矩阵，从而使得自然语言处理领域中的许多任务可以像图像任务一样使用 CNN 网络进行解决。

一个典型的卷积神经网络通常有三层：一个是卷积层，记为 Conv；一个是池化层，记为 Pool；一个是全连接层，记为 FC。一个最简单的卷积神经网络的结构如图 2-1 所示。虽然仅用卷积层也有可能构建出较好的神经网络，但大部分神经网络模型都会添加池化层和全连接层以使得网络抽取出的特征更加抽象。

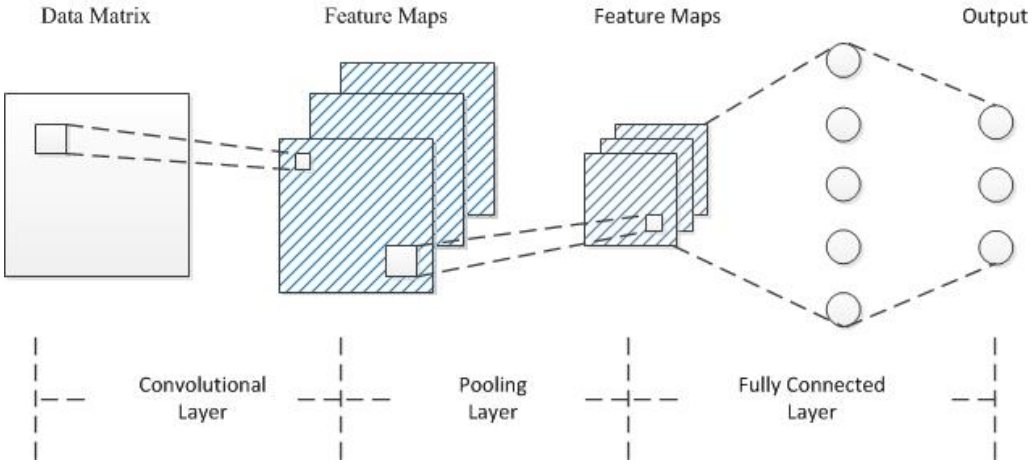

图 2-1 CNN 示意图

卷积层用于抽取数据矩阵（Data Matrix）的局部特征，通过将数据矩阵 D 与卷积核（Kernel）K 做卷积运算得到原数据的局部特征图（Feature Map）F。图 2-2 示意了特征图 F 中第 1 行第 1 列的值 F_{11} 的卷积运算过程，式 2-1 描述了特

征图中每个元素具体的运算公式，式中 m 和 n 定义了卷积核 K 的尺寸大小，m 和 n 的值由用户决定。

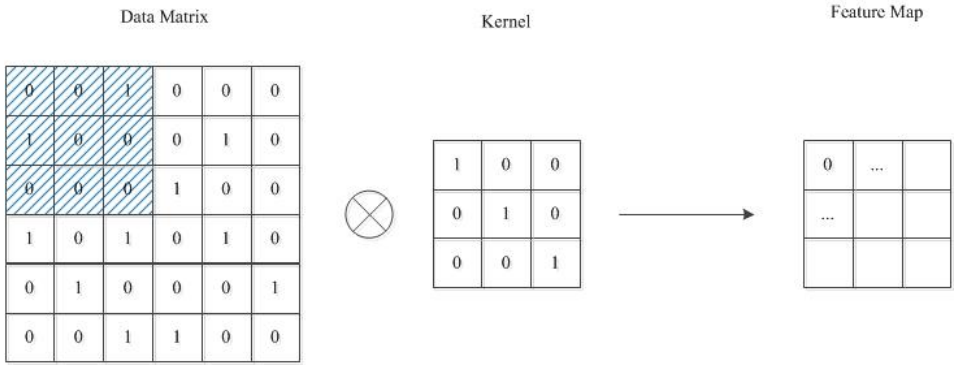

图 2-2 卷积过程示意图

$$F_{i,j} = \sum_{l=1}^{m} \sum_{w=1}^{n} D_{i-l+1,j-w+1} * K_{l,w} \tag{2-1}$$

池化层的作用是在不丢失数据原有信息的前提下，压缩局部特征图的尺寸，减少网络参数的数量，池化层通过对局部特征图进行降采样来达到这一目的。常用的降采样的方法有两种，一种是取池化窗口内的最大值作为输出的最大池化（Max Pooling），一种是取窗口内所有元素的平均值作为输出的平均池化（Average Pooling）。当窗口的尺寸设置为 $l*w$ 时，最大池化和平均池化的具体公式如式 2-2、2-3 所示。图 2-3 中的(a)、(b)分别示意了当窗口大小为 2*2 时对特征图进行最大池化和平均池化的结果。

$$P_{i,j} = max\left(F_{p+i,q+j}\right), 0 \le p \le l, 0 \le q \le w \tag{2-2}$$

$$P_{i,j} = \frac{\sum_{0 \le p \le l} \sum_{0 \le q \le w} F_{p+i,q+j}}{l*w} \tag{2-3}$$

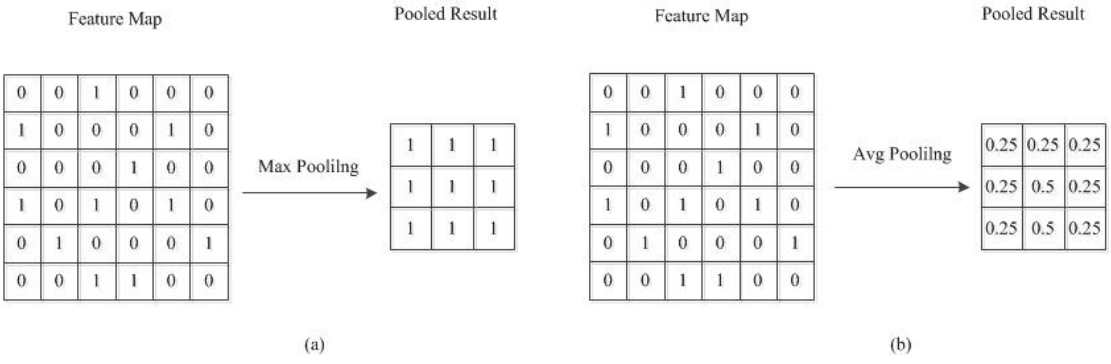

图 2-3 池化过程示意图

全连接层主要负责将上层网络中的神经元与下层网络中的神经元进行全部连接，虽然它可以对上层网络的输出结果进行降维，但却大大增加了网络中参数

的数量。若全连接层前面一层的输出结果不是一个向量，则需要先将其进行平铺，将其调整为一个向量，再进行全连接。此外，有时候会在全连接层中加入 Dropout 层，将上层中某些神经元随机丢弃（图中阴影区域），即不使其与下层神经元相连接，以防止网络过拟合。此过程示意图如图 2-4 所示。

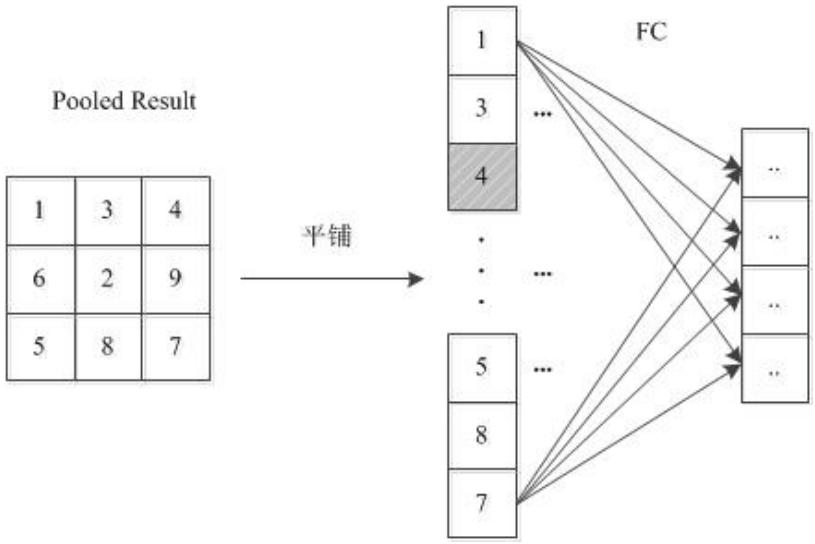

图 2-4 全连接过程示意图

除了上述核心的卷积层、池化层以及全连接层外，另外一个在 CNN 中被广泛应用到的是非线性激活函数，它们通常被用在各个层的结果输出之前，以提高模型的非线性表达能力。表 2-1 展示了常用的非线性激活函数的公式及优缺点。

表 2-1 常用激活函数及其优缺点

激活函数	计算公式	优缺点
sigmod	$sigmod(x) = \frac{1}{1+e^{-x}}$	优：是一种非线性函数； 缺：计算复杂，消耗资源；容易导致梯度消失和梯度饱和；模型拟合过程缓慢。
tanh	$tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	优：输出以零为中心； 缺：计算复杂；易导致梯度饱和。
ReLU	$ReLU(x) = max(x, 0)$	优：计算简单，速度快；基本上不存在梯度消失及梯度饱和，模型能够较快达到收敛； 缺：输出不以零为中心；可能会导致部分神经元无效，参数无法更新。
Leaky-ReLU	$Leaky-ReLU(x) = max(\alpha x, x)$	优：不存在梯度饱和的情况，模型可以快速达到收敛；神经元不会死亡； 缺：α 值需要人工设定。

在进行一些比较困难的任务时，上述简单的卷积神经网络往往达不到令人满意的效果，这时就需要构建一些比较复杂的网络结构。比较常用的一种模式是，将一个或多个卷积层后面跟一个池化层构造成一个模块，使用多个这种模块，然后后面依次连接多个全连接层，最后是 Softmax 层。业界熟知的 LeNet-5 网络、AlexNet 网络、VGG 网络等都是这种模式的应用。

2.1.2 RNN 及其变体 LSTM

传统神经网络模型中各个层之间是相连接的，但每层的各节点之间却并没有连接，所以对于时序问题，如想要预测某个句子下一个单词是什么，这类模型往往无能为力，因为想要预测某个单词，就必须需要与该单词相关的信息，而一个句子中的前后单词往往是相互关联的，所以就需要用到该单词周围的单词，但传统神经网络做不到这一点。为了解决这个问题，RNN 网络重用分类器，一个分类器用来进行状态总结，其他分类器接受对应时间步骤的训练并且传递状态，从而在避免需要大量历史分类器的前提下，使得当前节点与该层其他节点相连接。

不同于 CNN 网络，RNN 网络属于一种反馈网络，它会记住之前的信息并将其应用在当前输出的计算中，也就是说，隐藏层的输入同时包括输入层的内容和隐藏层上一时刻的输出内容。RNN 的结构图如图 2-5 所示。

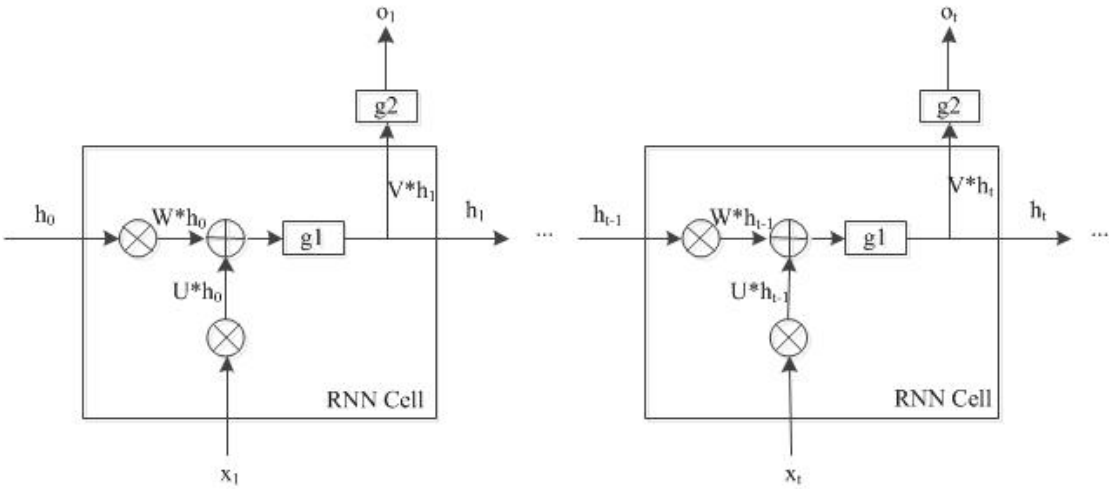

图 2-5 RNN 结构

RNN 的每个细胞单元包含输入单元、输出单元和隐藏单元，其中{x_1, x_2,…, x_{Tx}}代表输入序列，{h_1, h_2,…, h_{Tx}}代表隐藏层状态序列，{o_1, o_2,..., o_{Ty}}代表输出序列。h_t 和 o_t 分别代表第 t 时刻的隐藏状态和输出值，其各自的计算公式如下所示。

$$h_t = g1(U * x_t + W * h_{t-1}) \tag{2-4}$$

$$o_t = g2(V * h_t) \tag{2-5}$$

上式中的 *W*，*U*，*V* 均为网络中的待学习参数，*g1(.)*和 *g2(.)*都是非线性激活函数，*g1(.)*一般为 *tanh* 或 *ReLU*，*g2 (.)*的选取取决于模型所解决的问题，如果是一个二分类问题，*g2(.)*一般使用 *sigmoid* 激活函数；若是一个多分类问题，*g2(.)*一般使用 *softmax* 激活函数。

RNN 中每个时间步的输出并不是必须的，常用的 RNN 网络有四种类型：分类问题所使用的“多对一”（many-to-one），如图 2-6 中的(a)；音乐生成任务所使用的”一对多”（one-to-many），如图 2-6 中的(b)；命名实体识别任务所使用的“多对多”（many-to-many），如图 2-6 中的(c)；机器翻译任务所使用的“多对多”，如图 2-6 中的(d)。

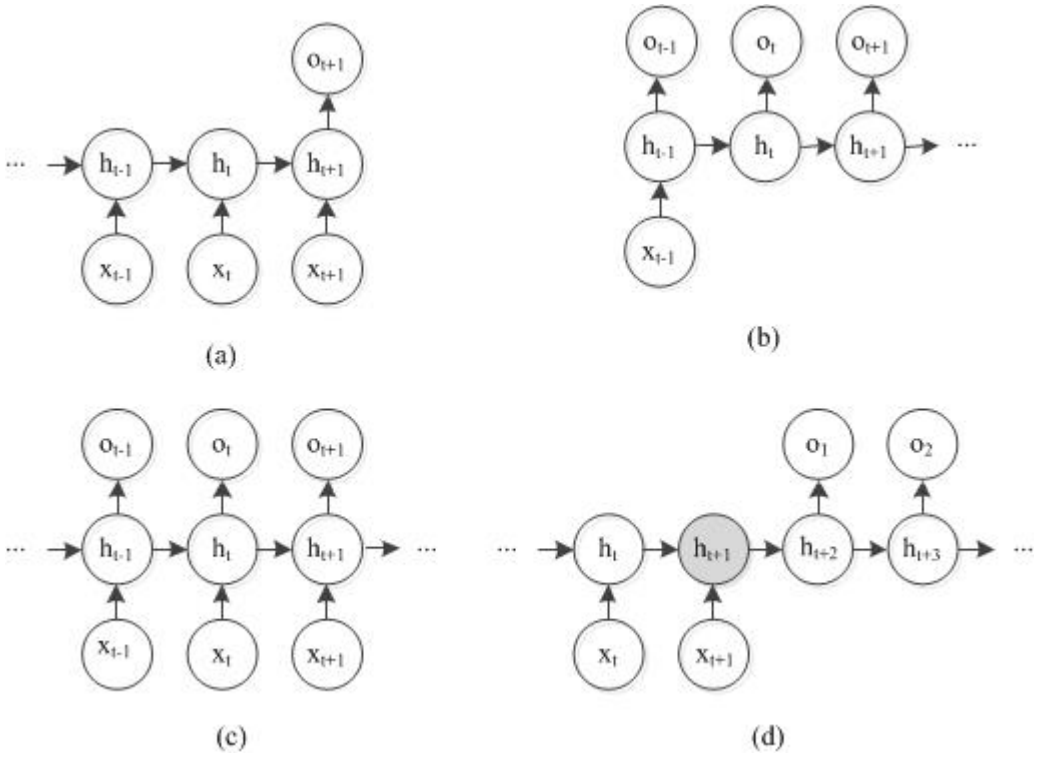

图 2-6 常用 RNN 网络类型

虽然理论上 RNN 可以保存任意长的历史信息来辅助当前时间点网络的决策，然而由于在反向传播过程中，会发生梯度消失或者梯度爆炸，导致梯度无法准确合理的传播到比较靠前的时间点。因此， RNN 虽然比前馈网络记住的历史信息要长，但实际上，它并无法记住非常靠前的历史信息。

解决上述问题一个很简单的方法是进行梯度截断，即在优化 RNN 网络参数的时候，给梯度设置一个上限值和一个下限值，当梯度小于下限值时，就将其调整为此下限值，上限值同理。但是这样直接通过上下限来调整梯度会导致梯度传播的不准确，所以长短时记忆网络 LSTM 被提出，其结构图如图 2-7 所示。

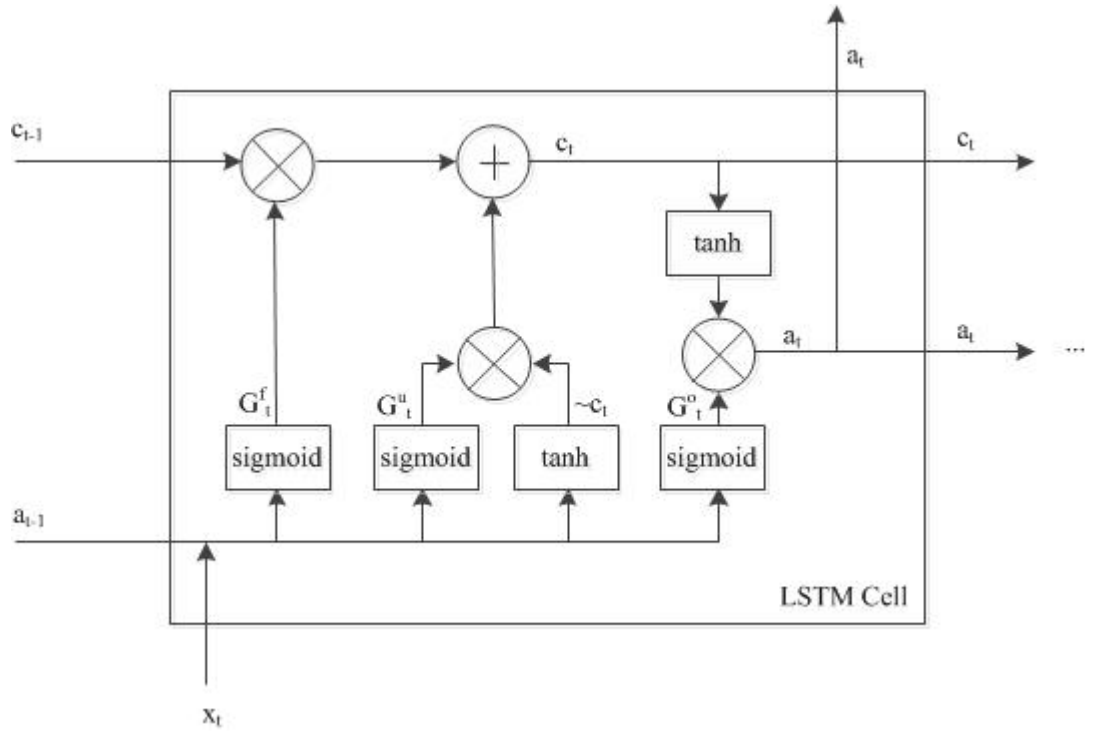

图 2-7 LSTM 结构图

图中 x 为输入向量；c 为细胞状态向量，它可以使梯度在网络中无损传播；a 为隐藏层状态向量，它将简单的反馈结构升级为模糊历史记忆的结构；G^f_t 为“遗忘门”，目的是解决新信息装载过多而导致激活函数饱和的问题；G^u_t 为“输入门”，它判断输入 x 中有哪些信息需要被编码并把新信息平稳安全的引入到细胞状态向量中；G^o_t 为“输出门”，它是为了让网络能选择合适的记忆内容进行输出而设置的。图中各个变量的计算方式如式 2-6 至 2-9 所示，其中 G^u_t 和 G^f_t 与 G^f_t 的计算类似，只是参数变量不一样，即变量不再是 W_f 和 b_f，所以不再赘述。

$$\mathrm{G}_t^f = \sigma\left(W_f[a_{t-1}, x_t] + b_f\right) \tag{2-6}$$

$$\sim c_t = \tanh(W_c[a_{t-1}, x_t] + b_c) \tag{2-7}$$

$$c_t = \mathrm{G}_t^f * c_{t-1} + \mathrm{G}_t^u * \sim c_t \tag{2-8}$$

$$a_t = \mathrm{G}_t^o * \tanh(c_t) \tag{2-9}$$

式中 W_f，b_f，W_c，b_c 均为网络中的待学习参数，$[a_{t-1},x_t]$表示将 a_{t-1} 和 x_t 进行拼接，$\sigma(.)$为 *sigmoid* 激活函数。

2.1.3 双向 RNN

单向循环神经网络的一个最主要问题是，它们只从之前的时间步骤中学习表示，但有时我们还可能需要从未来的时间步骤中学习表示，以便更准确的理解上下文环境，消除文本歧义。比如，“He said, Teddy bears are on sale”和“He said, Teddy Roosevelt was a great President”，在上面两句话中，当看到“Teddy”和它前面的两

个词“He said”时，并不能理解“Teddy”指的是“President”还是“Teddy bears”。所以，为了解决这种歧义性，需要获得“未来时刻”的文本信息，此即双向循环神经网络所能实现的。双向循环神经网络的结构图如图 2-8 所示。

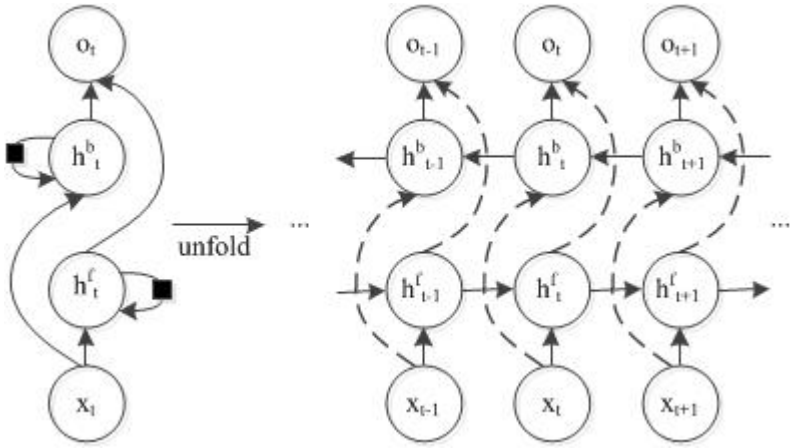

图 2-8 双向 RNN 结构图

x_t 表示第 t 时刻的输入值，o_t 表示第 t 时刻的输出值，h_t^f、h_t^b 分别表示第 t 时刻从左到右序列和从右到左序列的隐藏状态值。当网络中的隐藏层单元使用基本的 RNN 时，网络中各变量的计算方式如式 2-10 至 2-12 所示。

$$h_t^f = g1(U^f * x_t + W^f * h_{t-1}^f) \tag{2-10}$$

$$h_t^b = g1(U^b * x_t + W^b * h_{t-1}^b) \tag{2-11}$$

$$o_t = g2(V * c(h_{t-1}^f, h_{t-1}^b)) \tag{2-12}$$

U^f，W^f，U^b，W^b，V 都是待学习参数，c 是将 h^f_t 和 h^b_t 中所携带的信息进行融合的操作，可以简单的将其进行拼接，也可以应用前馈神经网络来实现。*g1(.)*，*g2(.)*的含义与式 2-4、2-5 中一致。此外，隐藏层单元也可以使用 LSTM 和 GRU 等 RNN 的变体。

2.2 基于 Encoder-Decoder 框架的 Seq2Seq 模型

Encoder-Decoder 即编解码架构，它的核心思想是将一个现实问题转化为一个数学问题，然后通过求解这个数学问题从而达到解决现实问题的目的。Encoder-Decoder 的结构图如图 2-9 所示。

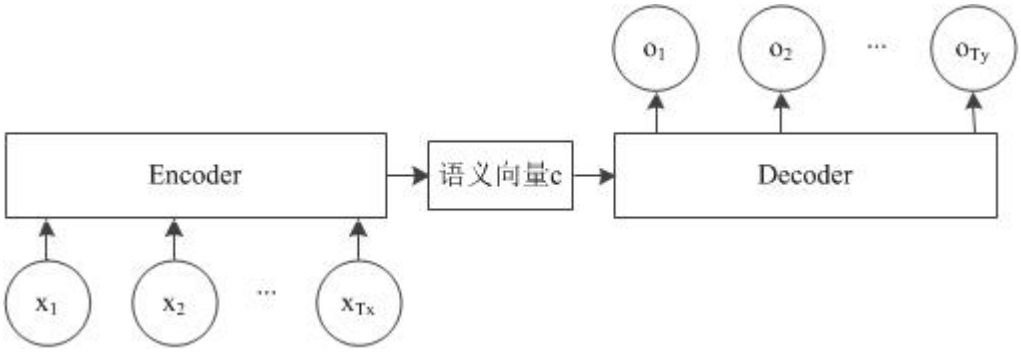

图 2-9 Encoder-Decoder 框架

图 2-9 中的 Encoder 即编码器，它的作用是将现实问题转化为数学问题，即将输入内容转化为一个向量，输入内容可以是图片、文字或者音频等；Decoder 即解码器，它的作用是求解上述数学问题，从而找到解决现实问题的方案，即根据编码器的输出向量进行解码得到输出，此输出可以是音频或者文字等。

在 Encoder-Decoder 模型架构中，编码器（Encoder）和解码器（Decoder）的实现方式都很灵活，深度学习中常用的 CNN、RNN 等都可以使用，但对于序列到序列问题而言，即根据一个输入序列得到其相对应的输出序列，它要求模型中的编码器和解码器能够对序列进行较好的处理，所以此时较常使用基于 RNN 的 Encoder-Decoder 来构建 Seq2Seq 模型，当使用双向 RNN 构建编码器，使用单向 RNN 构建解码器时，模型的结构如图 2-10 所示。

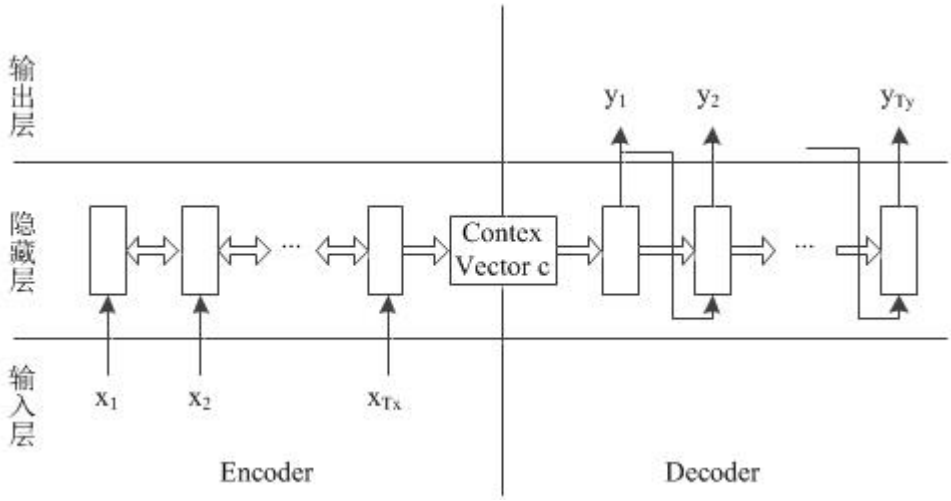

图 2-10 Seq2Seq 模型

对于一个数据序列对<*src*,*tgt*>而言，模型中的输入序列{$x_1, x_2,\ldots, x_{Tx}$}即原文本 *src*， T_x 表示 src 中的词语数，输出序列{$y_1, y_2,\ldots, y_{Ty}$}即标准摘要 *tgt*，T_y 表示 *tgt* 中的词语数。在编码阶段，输入序列中的每一个元素首先会被编码为一个固定大小的词向量，然后隐藏层将各单元携带的信息进行相互传递后得到每个单元

的隐藏层状态值，最后得到输入序列的语义向量表示。语义向量 c 的计算公式如式 2-13 所示。

$$c = q(\{h_1, h_2, \dots, h_{T_x}\}) \tag{2-13}$$

式 2-15 中 h_t 表示编码器第 t 时刻的隐藏层状态值，$q(.)$表示非线性操作，则向量 c 是由隐藏状态序列经过非线性操作得到的，此非线性操作可以通过直接使用编码器中最后一个时刻的隐藏层状态 h_{Tx} 来实现，也可以通过对隐藏层状态集合加权求和来实现。h_t 的计算方式 2-14 所示。

$$h_t = f(x_t, h_{t-1}) \tag{2-14}$$

即第 t 时刻隐藏层状态值是在第 t-1 时刻隐藏层状态值及第 t 时刻输入值的共同作用下产生的。

解码器在解码的第 t 时刻要根据编码器生成的文本向量表示 c 以及之前所有时刻预测得到词语$\{y_1, y_2, \dots, y_{t'-1}\}$的条件下，得到第 t'时刻预测词语 $y_{t'}$的概率分布：

$$p(y_{t'}|\{y_1, y_2, \dots, y_{t'-1}\}; c) = g(y_{t'-1}, s_{t'}, c) \tag{2-15}$$

其中 $g(.)$为非线性操作，$s_{t'}$表示解码器在解码第 t'时刻隐藏层的状态值，其计算方式如式 2-16 所示。

$$s_{t'} = f(y_{t'-1}, s_{t'-1}, c) \tag{2-16}$$

即第 t'时刻解码器隐藏层状态值是在解码器第 t'-1 时刻的生成词、第 t'-1 时刻的隐藏层状态值及输入序列向量表示的共同作用下产生的。式中 $f(.)$代表非线性操作。

最终，在解码器获得每一时刻的预测概率分布后，模型将目标函数由联合概率转化为有序的条件概率之积，即：

$$\begin{aligned} P\left(y_1, y_2, \dots, y_{T_y} \middle| x_1, x_2, \dots, x_{T_x}\right) &= \prod_{t'=1}^{Ty} p(y_{t'}|\{y_1, y_2, \dots, y_{t'-1}\}; c) \\ &= \prod_{t'=1}^{Ty} g(y_{t'-1}, s_{t'}, c) \end{aligned} \tag{2-17}$$

2.3 集束搜索

对于序列生成任务而言，如文本摘要，解码器在解码的每一步都需要使用相关的搜索算法来确定当前解码时刻所预测得到的词语。解码每一步确定当前预测词语最简单的做法就是选择当前预测概率最大的词语，但是这属于一种贪心的做法，虽然它可以保证每一步预测得到的词语是当前最优的，但是如此获得的最终生成序列却不能保证最优。维比特算法是针对篱笆图提出的一种动态规划算法，它可以保证找出图中的最短路径，但是当图中节点的后继节点很多时，此算法效率较低。比如文本摘要，每一个解码时刻预测得到的概率分布中都包含几万个词

语。

集束搜索（Beam Search）尝试在广度优先的基础上进行搜索空间和时间的优化，即在解码的每一步保留一定数量（即集束宽度 *beam_size*）的最优解，从而使得最终的解与全局的最优解较为接近。假设系统词汇表大小为 V，生成序列长度为 L，集束宽度记为 B，则算法过程如下：

1. 设置 t=1，从解码的第一时刻预测得到的概率分布中挑选概率最大的 B 个词语作为当前预测结果$s^t = \{s_1, s_2, \dots, s_B\}$，记录 s^t 中每个序列的生成概率。此时 s^t 中含有 B 个序列，每个序列 s_i 中只含有一个词语；
2. t=t+1，将当前生成结果 s^{t-1} 中每个序列的生成概率与解码第 t 时刻预测到的概率分布进行组合，然后从这 B*V 个结果中选择联合概率最大的 B 个作为第 t 时刻的生成结果 s^t，记录 s^t 中每个序列的生成概率。此时 s^t 中含有 B 个序列，每个序列长度为 t。
3. 若 t<L，转至第 2 步，否则算法结束。

2.4 本章小结

本章最开始对本文文本分类研究中所涉及到的基础模型 CNN 进行了介绍，随后对序列到序列模型的编解码器中常用到的 RNN、LSTM 及双向 RNN 进行了介绍，最后对生成式文本摘要所使用的 Seq2Seq 模型及解码搜索算法—集束搜索进行了介绍。

第三章 文本摘要技术研究

本章首先对本文所使用数据集中存在的问题进行了数据处理方式解释，并对构建的带注意力机制的 Seq2Seq 模型进行了说明，然后对本文所提出的三个文本摘要模型进行了详细的介绍。这三个模型分别是：为了使模型对文本中关键内容更加敏感所提出的基于关键词调整注意力机制的生成式模型；针对使用词级别编码器语义表达不充分的问题提出的一种双编码器摘要模型；针对原文中部分内容冗余问题提出的一种双阶段式文本摘要模型。

3.1 数据预处理

观察数据集中的原文内容发现，文本中存在大量噪声信息，如“…浏览器不支持video标签”、“显示图片”等，这些内容严重扰乱了文本信息的表达，使得数据集的质量下降。为此，本文首先对数据进行了过滤操作，提高原文的可读性和简洁性。此外，文本中的网址、邮箱、电话号码等，很显然对于标准摘要的生成是无用的，但是又不能将其直接删除，否则会破坏语言的逻辑性和完整性，所以本文将其都替换为统一的符号。每一种数据清洗的具体操作如表3-1所示。

表 3-1 数据清洗操作

文本内容	操作
……，北京电视台科教频道播出的《法制进行时》节目，再次将暴恐动漫提出来进行批评。具体报道可以快进到 22:44 秒。{!--<Paragraph>PGC_VIDEO:{\"mp4_url\":<Paragraph>\"http://video.proc.sina.cn/video_explore/location.php?video_id=137840610\"}<Paragraph>--}您的浏览器不支持 video 标签。	将左侧文本中灰色部分删除
“……不可以找来第三势力对抗马英九，这是极度不理智的行为。环球网免责声明版权作品，未经环球网 Huanqiu.com 书面授权，严禁转载，违者将追究法律责任。”	将左侧文本中灰色部分替换为“TAG_wz”
……。■记者戴鹏（原标题：安化最牛违建 6 道停工令叫不停已建 32 层将公开预售）来源： http://news.sina.com.cn/c/2014-11-25/110831198751.shtm	将左侧文本中灰色部分替换为“TAG_wz”
……，欢迎关注财视 Media 微信公众号：财视 Media(ID:caishibagua)更多精彩内容，企业八卦，等你来挖。加入我们的 Q 群：418295218 让我们一起八卦	将左侧文本中灰色部分替换为“TAG_nb”

3.2 加入注意力机制的 SeqSeq 文本摘要模型

传统的Seq2Seq模型中解码器在预测不同的输出时所使用的语义向量是不变的，因为编码器把输入序列编码为一个固定的向量，这显然不合理。且随着输入序列的增长，固定语义向量所含的信息越来越有限，导致模型对原文语义理解不准确，模型效果差[10]。基于这一问题，Bahdanau在机器翻译任务中提出了注意力机制。

通过加入注意力机制，可以减轻传统Seq2Seq模型中上下文向量c的信息负担，使得模型在进行解码时，解码器对输入序列中各个元素可以有不同的关注度，进而生成更加准确的摘要内容。为此，本文使用加入了注意力机制的Seq2Seq模型来构建文本摘要模型，模型中编码器使用双向LSTM来构建，解码器使用单向LSTM来构建，模型结构如图3-1所示。此时解码器每一个时刻预测得到的概率分布的计算公式由式2-17变为式3-1。

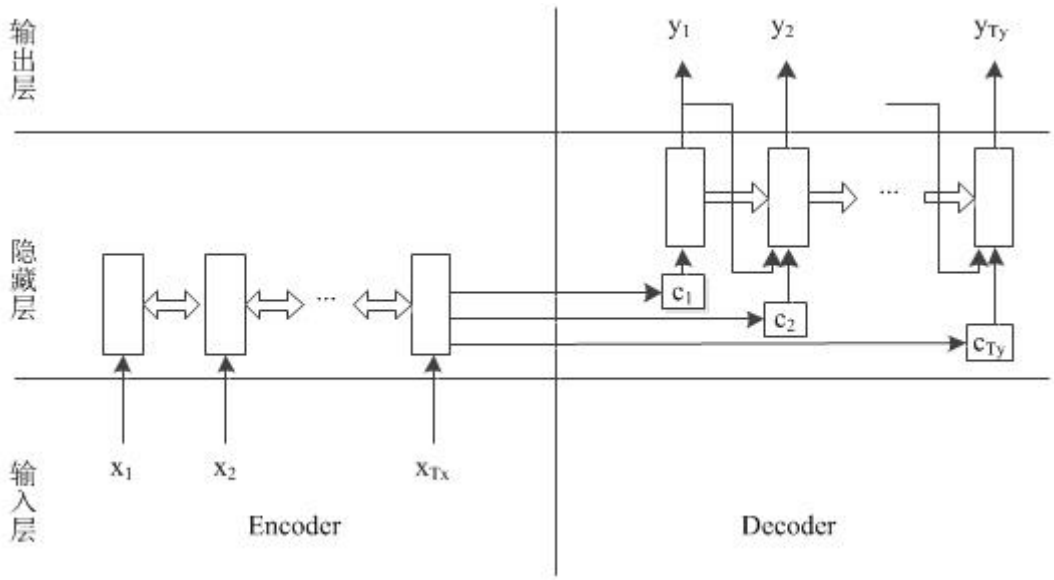

图 3-1 加入注意力机制的 Seq2Seq 模型

$$P(y_{t'}|\{y_1, y_2, \ldots, y_{t'-1}\}; c_{t'}) = g(y_{t'-1}, s_{t'}, c_{t'}) \tag{3-1}$$

由式 3-1 可知，此时每个解码时间步的预测概率分布不再取决于固定变量 c，而是取决于目标单词 $s_{t'}$的环境语义向量 $c_{t'}$。语义向量 $c_{t'}$取决于编码器中的隐藏层状态序列$\{h_1, h_2, \ldots, h_{Tx}\}$，解码第 t'时刻的变量 $c_{t'}$的计算方式如式 3-2 所示。

$$c_{t'} = \sum_{t=1}^{Tx} \alpha_{t't} h_t \tag{3-2}$$

由式3-2可知，$c_{t'}$是由隐藏层状态序列$\{h_1, h_2, \ldots, h_{Tx}\}$加权求和得到的，$h_t$表示编码器第$t$个单元的隐藏状态值，其中状态$h_t$的权重$\alpha_{t't}$表示原文中第$t$个词语在解码第$t'$时刻的重要性，其计算方式如式3-3所示。

$$\alpha_{t't} = \frac{\exp(e_{t't})}{\sum_{i=1}^{Tx} \exp(e_{t'i})} \tag{3-3}$$

$$e_{t't} = a(s_{t'-1}, h_t) \tag{3-4}$$

由式3-3可知，$\alpha_{t't}$是对$e_{t't}$进行归一化得到的值。在机器翻译中， a被视为一个对齐模型，用以获得$s_{t'-1}$和h_t的相关性，相关性的计算方式有拼接、点积等。当使用感知机来实现对齐模型时，$e_{t't}$的计算方法如式3-5所示。

$$e_{t't} = v^T tanh(W_s s_{t'} + W_h h_t + b_{attn}) \tag{3-5}$$

其中v，W_s，W_h，b_{attn}是网络需要学习的参数，v^T表示对v进行转置。$\alpha_{t't}$表示目标单词$y_{t'}$与原文单词x_t对齐的概率。因为感知机允许损失函数的梯度进行反向传播，所以对齐模型可以与模型中的其他模块一起被训练。在文本摘要任务中，$\alpha_{t't}$表示原文中的第t个单词x_t对预测目标生成单词$y_{t'}$所做的贡献，也就是说，在解码的第t'时刻，被预测的某个单词只与原文本中的某些输入单元具有较大的相关性，且各个输入单元的相关性会随着所被预测单词的不同而改变。

虽然注意力机制可以减轻传统Seq2Seq模型中上下文语义向量c的信息负担，使得模型在进行预测时，解码器对输入序列中各个元素可以有不同的关注度，进而生成更加准确的摘要内容。但是Tu等人[64]在进行机器翻译任务时发现，这种机制经常会导致“过度翻译”（over-translation）问题和“漏翻译”(under-translation)问题，“过度翻译”和“漏翻译”的释义及举例如表3-2所示，图3-2展示了一个中-英翻译的示例。

表 3-2 “过度翻译”和“漏翻译”

概念	释义	举例
过度翻译（over-translation）	在翻译的某些时候，原文本中的许多词语过度的被模型所关注和理解。	如图 3-2 中的“关闭”一词
漏翻译(under-translation)	在翻译的某些时候，原文本中的某些词语被忽略了，没有被模型考虑。	如图 3-2 中的“被迫”一词

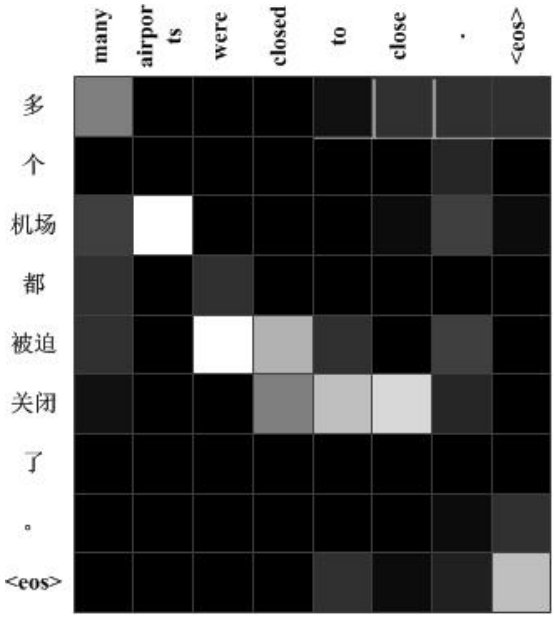

图 3-2 中-英翻译示例

这个问题产生的主要原因是，在解码器进行解码的每一步，注意力机制只能记录Encoder中每个词语在当前时刻的贡献度，而不能记录其历史贡献度，也就是说，在加入了注意力机制的序列到序列模型中，当前时刻的注意力与此前时刻的注意力是相互独立的，当前时刻的解码并不能从之前的操作中获取任何与对齐相关的信息，所以导致了模型出现上述两个问题。为此，本文参考See等人[17]的做法，在本文的摘要模型中使用了Coverage机制。

Coverage机制的主要思想是在模型中维护一个覆盖向量c_v用以记录编码器中各个输入单元的历史注意力信息，然后基于此信息，在解码的每一步，抑制编码器中那些在此前时刻对语义向量的生成贡献较多的隐藏层单元的贡献度。在解码的第时刻，覆盖向量$c_v_{t'}$由公式3-6所示的方式获得。

$$c\_v_{t'} = \sum_{j=1}^{t'-1} \alpha_j \tag{3-6}$$

式3-6中向量$\alpha_j=[\alpha_{j1},\alpha_{j2},\ldots,\alpha_{jTx}]$表示解码的第$j$时刻编码器中各隐藏层单元的注意力分布，向量中每个维度的值即根据式3-3计算得到的结果。$c_v_{t'}$即解码第t'时刻之前所有解码时刻注意力分布的总和，直观上可以将$c_v_{t'}$看做关于原文本中所有单词的一个分布，它表示到t'时刻之前，编码器中的各隐层单元在注意力机制中所做的贡献。向量c_v的初始值为0向量。

由于加入了覆盖向量，则在解码第t'时刻注意力的计算方式也要做相应的调整，即把式3-5调整为式3-7，$\alpha_{t't}$的计算方式，即式3-3不变。

$$e_{t't} = v^T tanh(W_s s_{t'} + W_h h_t + w_{c\_v} c\_v_{t't} + b_{attn}) \tag{3-7}$$

w_{c_v}是模型中的待学习参数。$c_v_{t'}t$表示向量$c_v_{t'}$中的第t个维度的值。在$e_{t't}$的计算中考虑$c_v_{t'}t$可以使得之前时刻的解码操作对当前时刻的注意力分布做相应调整，避免注意力模型重复的对同一个位置的输入内容进行过多的关注，从而避免生成的摘要中存在较多的重复内容，造成生成的文本内容不通畅。

3.3 基于关键词调整注意力机制的文本摘要模型

传统Seq2Seq模型的编码器在对输入序列进行语义理解的时候，会对输入序列中的各个单元不加选择的进行编码，即解码的每个时间步所使用的语境向量都是相同的，这对于机器翻译任务或者文本生成任务而言显然是不合理的，因为它将原文中的每个单词对于当前预测词语的重要性视作相同。此外，随着输入序列的增长，使用RNN或其变体对原文进行编码时，文本靠前位置的信息会被丢失，从而导致模型的效果快速下降[10]。借鉴Bahdanau等人[11]在翻译任务中提出的注意力机制，See等人[17]在其研究中使用了这种机制，从而使得在解码的每一步编码器都可以编码到一个更加准确的语义向量。借鉴人类阅读时的“选择性阅读”习惯，即人类在阅读一篇文章时，首先会笼统的浏览全文，找出关键段落，然后在这些段落里找文本的中心语句。Zeng等人[65]提出了选择性编码，通过使用门控网络，对编码器中每个时刻的隐层单元输出值计算得到一个权重，从而使得模型可以选择性的对原文中的内容进行编码。此外，一些学者[33,66]将文本关键词信息加入到模型中以辅助编码器编码。

基于以上学者的研究，本文提出一种基于关键词调整注意力机制的文本摘要模型。首先使用有监督算法获得原文本中每个词语属于文本关键词的概率，并基于此概率在解码的每一步对注意力模型的输出进行调整，然后获得语义向量完成预测。此外，由于传统模型在预测目标词语时只能生成模型词汇表中的词语，此模型词汇表是由数据集文本中的高频词构成的，所以此词汇表的大小通常会有一定限制，因为词汇表太大会造成模型在预测词语时的效率和效果下降，词汇表太小会导致太多的OOV词语及罕见词，而原文本中通常会有一些人名和地点词等，它们的词频通常较低，但对于文本的描述又十分重要，不应该被模型识别为‘UNK’。所以本文使用了指针网络(Pointer-generator Network) [17]，在生成模型词汇表中的词语的基础上，加入复制原文中词语的能力来提高摘要生成的准确性，即将抽取式摘要所具备的抽取能力引入到了生成式文本摘要的学习中。模型的整体结构如图3-3所示。

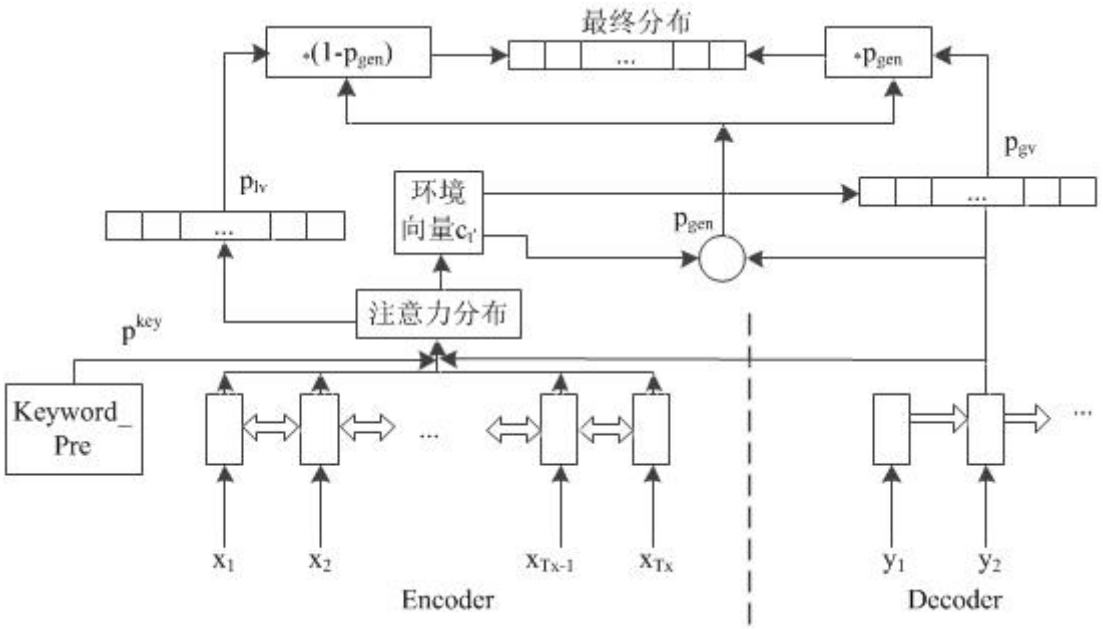

图 3-3 基于关键词改进注意力机制的文本摘要模型

图3-3中Keyword_Pre模块即上文所述的预测原文中每个词语属于关键词概率的模块，旨在通过每个词语属于关键词的概率调整在解码的每一时刻他们对上下文语义向量$c_{t'}$的贡献程度，使得模型对于原文中的关键词信息更加敏感，减弱非关键词对模型理解语义的影响。

对于文本摘要任务而言，标准摘要中的词语一定都是原文本中的关键词，而且本文最终需要获得的是原文中每个词语属于文本关键词的概率值。所以，对于Keyword_Pre模块，本文设计了一种关键词概率预测算法。首先使用一种无监督关键词提取算法获得原文的关键词集合，然后根据此集合中的词语对原文词语进行打标签，制作训练集，并按照序列标注的思想，使用有监督算法训练得到一个关键词概率预测模型，预测得到文本中每个词语属于关键词的概率。此过程示意图如图3-4所示，图中的“原文本集合”指本文所使用数据集中的所有原文本组成的集合，“摘要集合”指数据集中所有标准摘要组成的集合。

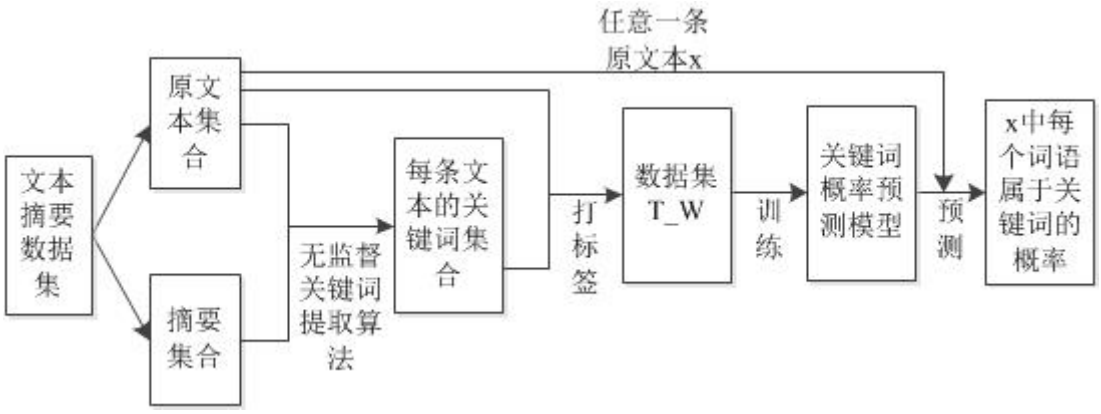

图 3-4 文本关键词概率预测过程

下面将分别对图3-4中的“无监督关键词提取算法”和“关键词概率预测模型”进行介绍。

1. 无监督关键词提取算法

TextRank算法源自于谷歌的PageRank算法[67]，相比于TF-IDF关键词提取算法它只需要利用单篇文档本身的信息即可提取出此篇文档的关键词，但它可能会不加选择的将属于大多数文档关键词集合中的词语选为此篇文档的关键词，算法不具有针对性。对于文本摘要任务而言，原文中属于标准摘要中的词语一定是原文的关键词，所以本文使用改进的TextRank算法来实现每条原文本的关键词提取。算法步骤如下：

1) 对原文和标准摘要进行分词、去停用词以及词性标注，然后将其中属于名词、数词、动词、形容词词性的单词保留下来，形成原文保留词集合和标准摘要保留词集合。使用原文保留词集合中的词语构成候选关键词集合 $C\_k=\{c\_k_1,c\_k_2,\ldots,c\_k_{Tx'}\}$，其中 $T_{x'}$是保留的候选关键词的数量，使用原文保留词集合和标准摘要保留词集合交集中的词语构成此篇文档的线索词集合。
2) 构建图模型 G=(V，E)，其中 V 是图中的节点集合，E 是图中的边集合。V 中每个节点对应候选关键词集合 C_k 中的某个候选词，节点的权重代表此词语的重要性。E 中每条边的权值即集合 C_k 中某两个候选词间的关联度。本文用词语间的共现关系来度量词语间的关联度。
3) 将线索词集合中词语所对应节点的初始权重设置为 2，其余节点的初始权重设置为 1，根据公式 3-8 对图中各个节点的权重迭代计算，直至图中各个节点的权重达到收敛。

$$WS(v_i)=(1-d)+d*\sum_{v_j\in Adj(v_i)}\frac{w_{ji}}{\sum_{v_k\in Adj(v_j)}w_{jk}}WS(v_j) \tag{3-8}$$

其中 $WS(v_i)$表示第 i 个节点的权重，d 是阻尼因子，经验值为 0.85。w_{ji} 表示词语 c_k_j 和词语 c_k_i 之间的关联度，Adj(v_i)是图中第 i 个节点的相邻节点所

构成的集合。

4) 从各节点最终计算得到的权重中选取出前 15 个权重最高的词语构成提取出的关键词集合 $Key=\{key_1,key_2,\ldots,key_{15}\}$。

2. 关键词概率预测模型

对于“关键词概率预测模型”，首先是制作训练集T_W，即图3-4中的“打标签”。

对于文本摘要数据集中的每一条原文$x=\{x_1,x_2,\ldots,x_{Tx}\}$，$x_t$表示原文中的第$t$个词语，对其按照上述改进的TextRank算法得到原文的关键词集合Key，然后将原文中属于Key中的词语标记为“1”，其余标记为“0”，从而得到原文的一个标签序列$x_t=\{x_t_1,x_t_2,\ldots,x_t_{Tx}\}$。其中$x_t_t\in\{0,1\}$，表示原文中第$t$个词语的标签，若$x_t\in Key$，则$x_t_t=1$，否则$x_t_t=0$。以此方法获得数据集T_W，T_W中的每条数据为$(x,x_t)$。然后使用此数据集按照序列标注的思想训练得到一个关键词概率预测模型。关键词概率预测模型由一层双向的LSTM实现，结构如图3-5所示。

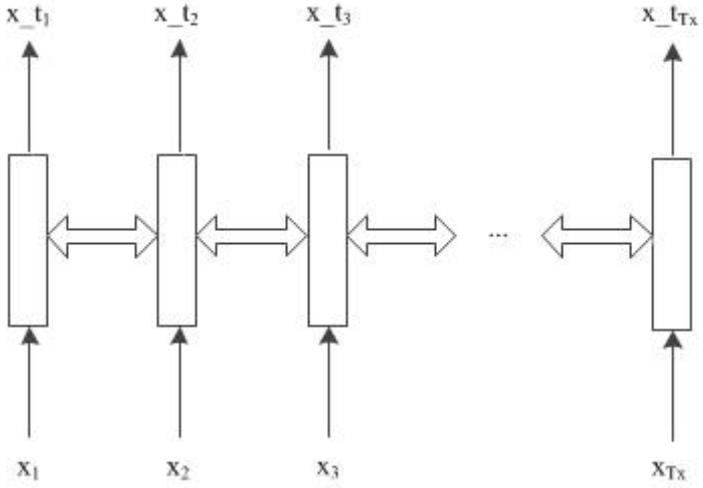

图 3-5 序列标注模型

模型训练完成之后，使用模型输出层的结果作为原文中每个词语属于关键词的概率，$p^{key}=[p^{key}_1,p^{key}_2,\ldots,p^{key}_{Tx}]$，其中$p^{key}_t$表示原文中第$t$个词语属于关键词的概率。

获得每个词语属于关键词的概率后，在解码的每一步，使用原文的关键词概率序列对编码器中每个单元获得的注意力进行调节，并获得解码第t'时刻的语义向量$c_{t'}$。

$$c_{t'}=\sum_{t=1}^{Tx}\alpha''_{t't}h_t \tag{3-9}$$

$$\alpha''_{t't}=\frac{\exp(\alpha'_{t't})}{\sum_{i=1}^{Tx}\exp(\alpha'_{t'i})} \tag{3-10}$$

$$\alpha'_{t't}=\alpha_{t't}*p_t^{key} \tag{3-11}$$

式3-11中的$\alpha_{t't}$即使用式3-3计算得到的结果，$\alpha'_{t't}$表示解码第t'时刻基于关键词调整之后编码器中第t个词语的重要性，随后使用归一化操作获得第t个词语对文本向量$c_{t'}$的贡献程度$\alpha''_{t't}$，再使用加权求和的方法获得在解码第t'时刻的语义向量$c_{t'}$。

此时覆盖向量$c_v_{t'}$的计算中所使用的是调整后的注意力，即式3-6调整为式3-12。

$$c\_v_{t'} = \sum_{j=1}^{t'-1} \alpha''_j \tag{3-12}$$

之后，指针网络根据解码第t'时刻的上下文向量$c_{t'}$计算得到待预测词语的生成概率p_{gen}，计算公式如式3-13所示。

$$p_{gen} = \sigma((w_c)^T c_{t'} + (w_s)^T s_{t'} + (w_y)^T y_{t'-1} + b_{gen}) \tag{3-13}$$

式中w_c和w_s、w_y、b_{gen}都是模型中的待学习参数，$s_{t'}$表示解码器第t'时刻的隐藏状态，$y_{t'}$-1表示第t'-1时刻解码器预测得到的词语。p_{gen}∈[0,1]。此时在解码的第t'时刻，解码器预测概率的计算公式如式3-14所示。

$$\mathrm{p}(\mathrm{y}_{t'}|\{y_1,\dots,\mathrm{y}_{t'-1}\};\{x_1,\dots,\mathrm{x}_{Tx}\}) = p_{gen} * p_{gv}(\mathrm{y}_{t'}) + (1 - p_{gen}) * p_{lv}(\mathrm{y}_{t'}) \tag{3-14}$$

对于每一个输入序列，此时模型在解码时最终所使用的是扩展词汇表，它包括模型词汇表中的所有词语以及所有出现在原文中的词语。p_{gen}可以看做是一个软开关，用于选择是生成模型词汇表中的词语，还是复制原文中的词语。$p_{gv}(y_{t'})$的计算公式如式3-15所示，代表模型词汇表中词语的概率分布。$p_{lv}(y_{t'})$ 的计算公式如式3-16所示，代表原文中与$y_{t'}$相同的词语的注意力权重之和。

$$p_{gv}(y_{t'}) = \mathrm{g}(y_{t'-1}; s_{t'}; c_{t'}) \tag{3-15}$$

$$p_{lv}(y_{t'}) = \sum_{t:x_t=y_{t'}} \alpha''_{t't} \tag{3-16}$$

如果词语$y_{t'}$是一个OOV词，则$p_{gv}(y_{t'})$值为0，如果它是原文本之外的词语，则$p_{lv}(y_{t'})$值为0。

3.4 双编码器文本摘要模型

加入了注意力机制的序列到序列模型，虽然减轻了固定环境向量的信息负担，使得解码器在解码的每一步都可以计算得到一个与当前隐藏状态更加相关的环境向量，但是由式3-3可以看出，实际上，环境向量$c_{t'}$是由词级别编码器中各个时刻的隐藏状态加权求和得到的结果，这就像早期获取一个句子的句向量采用将此句子中各个词语的词向量进行拼接或者直接相加的方式。当所使用的词向量表达的语义较浅时，这种由词向量获得文本语义向量，或者由词向量获得子句向量，再由子句向量获得文本语义向量的偏差便会较大。

为了解决上述问题，有些学者使用层级编码策略，如Nallapati[15]在编码器中设置了两个RNN模块，并在文本每个子句的末尾添加句末标志，低层次的RNN编码词项得到隐层状态后，将句尾标志对应的输出与其位置信息结合起来输入高层次RNN编码器进行子句级别的编码。本文参考有些学者在使用神经网络时，在嵌入层同时使用词级别向量和字符级别向量两个通道来增加网络的数据表达能力的思想，设计了一个具有两个编码器的生成式文本摘要模型，模型的整体结构如图3-6所示（为了简洁，图中未画出指针机制部分），两个编码器分别从较低层次的词语级别以及较高层次的子句级别对原文语义进行理解。此时，指针机制中p_{lv}的计算针对的是词级别编码器，且由于一个子句中存在多个关键词语，所以子句级别的编码器不使用Coverage机制。

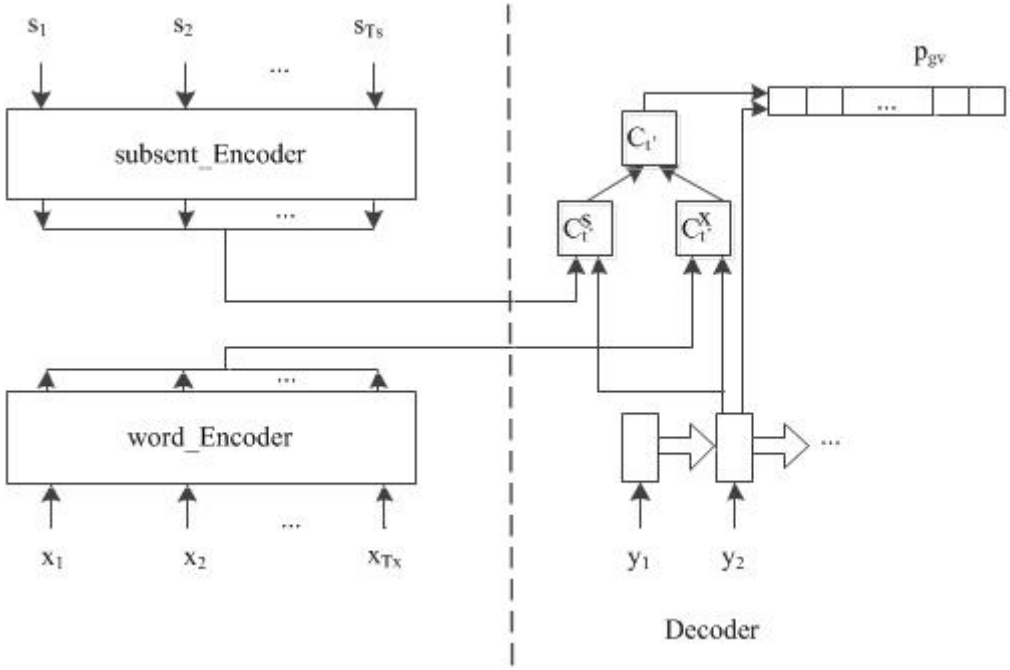

图 3-6 双编码器文本摘要模型

其中，word-Encoder表示使用词级别进行文本信息编码的编码器。本文使用双向LSTM实现，且嵌入层使用Word2vc词向量进行嵌入。word-Encoder的输入序列为$x=\{x_1,x_2,\ldots,x_{Tx}\}$，$x_t$表示文本序列中的第$t$个词语，输出序列为$h^x=\{h^x_1,h^x_2,\ldots,h^x_{Tx}\}$，$h^x_t$表示词级别编码器隐藏层第$t$时刻的输出值。$c^x_{t'}$表示在解码的第$t'$时刻，word-Encoder编码得到的上下文语义向量，其计算公式如式 3-17所示。

$$c^x_{t'} = \sum_{t=1}^{Tx} \alpha^x_{t't} h^x_t \tag{3-17}$$

其中，$\alpha^x_{t't}$表示解码第t'时刻，词级别编码器中第t个词语对语义向量$c^x_{t'}$的贡献程度，其计算过程如式3-18和3-19所示。

$$\alpha^x_{t't} = \frac{\exp(e^x_{t't})}{\sum_{i=1}^{Tx} \exp(e^x_{t'i})} \tag{3-18}$$

$$e^{x}_{t't} = (v^{x})^{T} tanh(W^{xs}s_{t'} + W^{xh}h^{x}_{t} + w_{c\_v}c\_v_{t't} + b^{x}_{attn}) \quad (3\text{-}19)$$

式中v^x，W^{xs}，W^{xh}，w_{c_v}，b^x_{attn}是词级别编码器中的待学习参数。此时$c_v_{t't}$所在向量$c_v_{t'}$的计算中使用的是解码第t'时刻之前各时刻词级别的注意力分布。

subsent-Encoder表示使用子句获得文本特征的编码器。考虑到BERT预训练模型强大的语言表征能力，所以此编码器的嵌入层直接使用由BERT获得的子句向量进行嵌入。

本文使用如下方式来借助BERT获得子句语义向量。对于原文本，先对其按字符进行切分，获得$c=\{c_1,c_2,\ldots,c_{Tc}\}$，$c$中的每个元素代表原文中的每个字符，$T_c$代表原文中的字符数量。然后按如下步骤获得文本各子句的语义向量：

a) 根据标点符号“。、？、！”将原文本其切分成子句，则此时原文表示为 $s=\{s_1,s_2,\ldots,s_{Ts}\}$。其中 T_s 是子句数，$s_k=\{c^k_1,c^k_2,\ldots,c^k_{Tck}\}$表示第$k$条子句，$c^k_m$ 表示第 k 条子句中的第 m 个字符，T_{ck} 是第k条子句的字符数量；

b) 在每条子句的末尾添加符号‘CLS’，并重新将子句整合为一条文本，即 $c'=\{c^1_1, c^1_2,\ldots, c^k_{Tck},\text{'CLS'}, c^{k+1}_1,\ldots,\text{'CLS'}\}$；

c) 将 c'输入到 BERT 网络中，最终通过输出‘CLS’所在位置的向量获得每个子句的语义向量 $s\_v=[s\_v_1,s\_v_2,\ldots,s\_v_{Ts}]$，$s\_v_k$ 表示文本中第 k 条子句的语义向量。

图3-7是一个含有2个子句、4个字符的文本序列使用BERT获得子句向量的示意图。其中s_v_1是输入序列中第一个‘CLS’符号对应位置的输出，s_v_2是输入序列中第二个‘CLS’符号对应位置的输出。

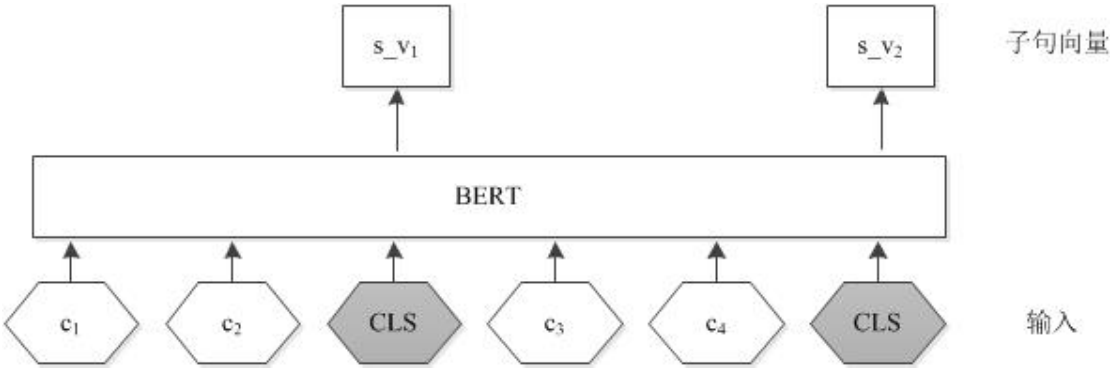

图 3-7 BERT 获得子句向量示意图

subsent-Encoder的输入序列为$s_v=\{s_v_1,s_v_2,\ldots,s_v_{Ts}\}$，输出序列为$h^s=\{h^s_1,h^s_2,\ldots,h^s_{Ts}\}$，$h^s_k$表示子句级别编码器隐藏层第$k$时刻的输出值。$c^s_{t'}$表示在解码的第$t'$时刻，subsent-Encoder编码得到的上下文语义向量，其计算公式如式3-20所示。

$$c^{s}_{t'} = \sum_{k=1}^{Ts} \alpha^{s}_{t'k} h^{s}_{k} \quad (3\text{-}20)$$

$\alpha^s_{t'k}$表示解码第t'时刻，子句级别编码器中第k个子句对语义向量$c^s_{t'}$的贡献程度，$\alpha^s_{t'k}$可由式3-21得到。

$$\alpha^s_{t'k} = \frac{\exp(e^s_{t'k})}{\sum_{i=1}^{Ts}\exp(e^s_{t'i})} \tag{3-21}$$

$$e^s_{t'k} = (v^s)^T tanh(W^{ss}s_{t'} + W^{sh}h^s_k + b^s_{attn}) \tag{3-22}$$

式中v^s，W^{ss}，W^{sh}，b^s_{attn}是子句级别编码器中的待学习参数。

最终在解码的第t'时刻，模型编码得到的上下文语义向量$c_{t'}$可由式3-23所示的方式得到。d_t是$c^x_{t'}$的权重，它可以像指针网络中的生成概率p_{gen}一样使用一个门控网络来获得，其具体计算方法如式3-24所示。

$$c_{t'} = d_t c^x_{t'} + (1 - d_t)c^s_{t'} \tag{3-23}$$

$$d_t = \sigma(w^{dx}c^x_{t'} + w^{ds}c^s_{t'} + b_d) \tag{3-24}$$

式3-24中的w^{dx}，w^{ds}，b_d是模型中的待学习参数。

获得了上下文向量后，剩余部分与加入了指针网络的计算过程一致。需要注意的是，由于subsent-Encoder是基于子句进行编码的，所以此时解码第t'时刻中p_{lv}的计算针对的是词级别注意力分布，即式3-16调整为如下：

$$p_{lv}(y_{t'}) = \sum_{t:x_t=y_{t'}} \alpha^x_{t't} \tag{3-25}$$

3.5 双阶段式文本摘要模型

本文通过对所使用数据集中的数据进行观察发现，许多平台在报道一个舆情事件的时候，往往会将与此事件有关的所有内容，甚至是人物生平信息等全部写入新闻中，但这些内容通常对于新闻标题的生成是无用的。此外，文本中还有一些内容属于冗余信息。两种情况的举例如表3-2中的数据所示，灰色部分属于无用和冗余信息。

表 3-2 无用和冗余信息示例

原文	4 月 27 日上午 10 时许，华商报记者在苏州中学校门口看到，学校所有学生正在操场跑操，而位于学校西侧的一栋教学楼下，警方拉起了警戒线，正在调查坠落原因。“我早上开门市时，有人议论苏州中学有学生坠楼了。”在苏州中学校门口经营一家店铺的女士说道。随后，华商报记者赶到榆林市星元医院急诊科，家属和几名学生焦急地站在抢救室门外。关于学生坠楼的原因，一位学生表示不清楚，其他情况也不愿多讲。事后，榆林市公安局榆阳分局崇文路派出所民警到苏州中学和星元医院进行调查。据现场调查民警介绍，坠楼学生为一名九年级(初三)男生，今年 16 岁，是从 5 楼坠落的。目前坠楼学生肺部出血，有几处骨折，暂无生命危险。事情发生后，教育局高度重视，立即成立事件调查小组奔赴现场，并上报区委、区政府，协调医院全力展开救治。对于该生坠楼的原因，目前，学校、区教育局和区公安机关正在进一步调查中。华商记者 张云飞陈冰编辑：华商报供稿榆林坠楼中学生确认为男生暂无生命危险华商网-华商报@2015-04-27,10:40:15
标准摘要	今日上午，榆林苏州中学一初三男生 坠楼，疑因感情 受挫，目前正在抢救中(图)
原文	……在杨卫泽之前，2014 年已有西宁的毛小兵、太原的陈川平、昆明的张田欣、广州的万庆良、济南的王敏等五位省会首府城市市委书记被调查或遭到解职。另外，十八大以来，已经有蒋洁敏、李东生、杨金山、令计划等四名十八届中央委员和李春城、王永春、万庆良、陈川平、潘逸阳、朱明国、王敏、杨卫泽等八名中央候补委员落马。■附：杨卫泽简历杨卫泽，男，汉族，1962 年 8 月生，江苏常州人。1988 年 4 月入党，1981 年 8 月参加工作。在职研究生学历，硕士学位。1978 年 12 月起，南京航务工程专科学校港口水工建筑专业学习；1981 年 8 月起，省交通厅规划计划处办事员、科员、基建计划科副科长、科长（其间：1986 年 5 月—1988 年 7 月挂职任邳县加口乡乡长助理）；……；2001 年 1 月起，苏州市委副书记、代市长、市长（其间：2004 年 6 月—2004 年 9 月参加中组部赴哈佛大学公共管理高级人才培训班学习）；2004 年 11 月起，无锡市委书记；2006 年 11 月起，省委常委、无锡市委书记；2011 年 3 月起，省委常委、南京市委书记
标准摘要	南京市委书记杨卫泽被查，传其攀附周永康；其四位搭档全部落马，曾被戏称\"近淤泥而不染\"；被查或与某退休高官实名举报有关。

如果将这些冗余信息连同关键信息一起输入到生成式文本摘要模型中进行训练，一方面，这些信息会对文本主题的表达造成混乱，使得模型编码器编码得到的语义向量对文本主题的表征有偏差；另一方面，这些信息增加了模型的输入长

度，且Seq2Seq模型中的编码器经常使用RNN及其变体来实现，它们较适用于短序列，过长的序列一方面会导致文本信息在编码的过程中逐步丢失，另一方面会导致梯度在反向传播的过程中逐渐消失，则此时模型的效果会受到影响。当按词语数进行统计时，本文所使用数据集的原文本长度从18到13918不等，长度分布如图3-8所示。

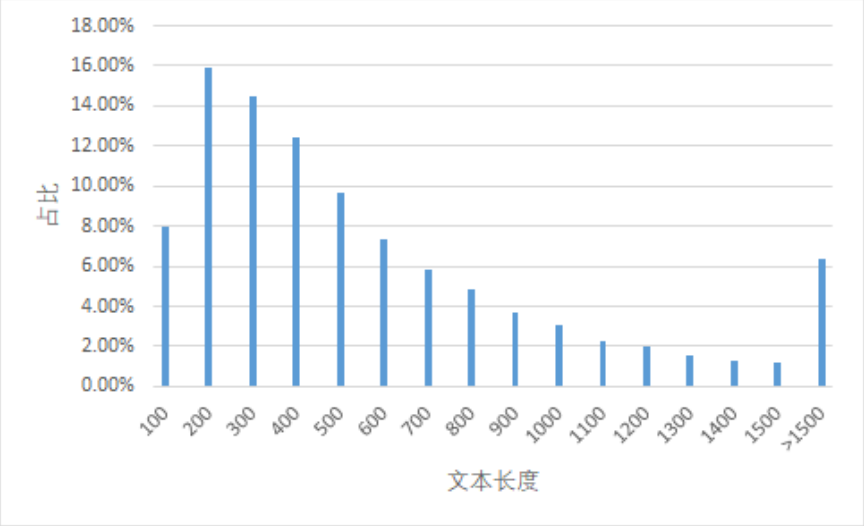

图 3-8 数据集原文本长度分布

图 3-9 中横坐标为 200 对应的一项代表文本长度大于 100 且小于 200 的文本数占数据集中总文本数的比例，其余类似，最后一项是指长度大于 1500 的文本数占数据集中总文本数的比例。由图可以看出，长度大于 400 的文本占到了文本总数的 50%。

另外，虽然从建模的角度来看，端到端模型具有吸引力，然而，有证据表明，当人们进行概括时，遵循两步法：首先从原文中选择出重要的短语或子句，然后再对它们进行进一步的释义[68]。在图像字幕中也有类似的证据，Anderson 等人[69]提出了一种双阶段模型，这个模型首先对待切割的目标物体预先计算得到一个边界框，然后再在这些区域内运用注意力机制来进行进一步的计算。

基于以上分析，本文提出了一种双阶段文本摘要模型，先从原文本中将与文本主题最为相关的子句抽取出，且尽可能的保证这些子句中包含更多文本关键词以及标准摘要中的词语，然后将抽取出的内容作为生成式文本摘要模型的输入，进行第二阶段的学习和训练。第二阶段使用的是带注意力机制、Coverage 机制和指针网络的生成式模型。

由于本文所使用的数据集是生成式文本摘要的公开数据集，所以对于此数据集而言，双阶段文本摘要模型第一阶段所需要抽取出的文本内容并没有一个实际的参考，于是本文采用一种启发式方法，选取出原文中的重要内容作为第二阶段模型的输入。此部分过程的示意图如图 3-9 所示。

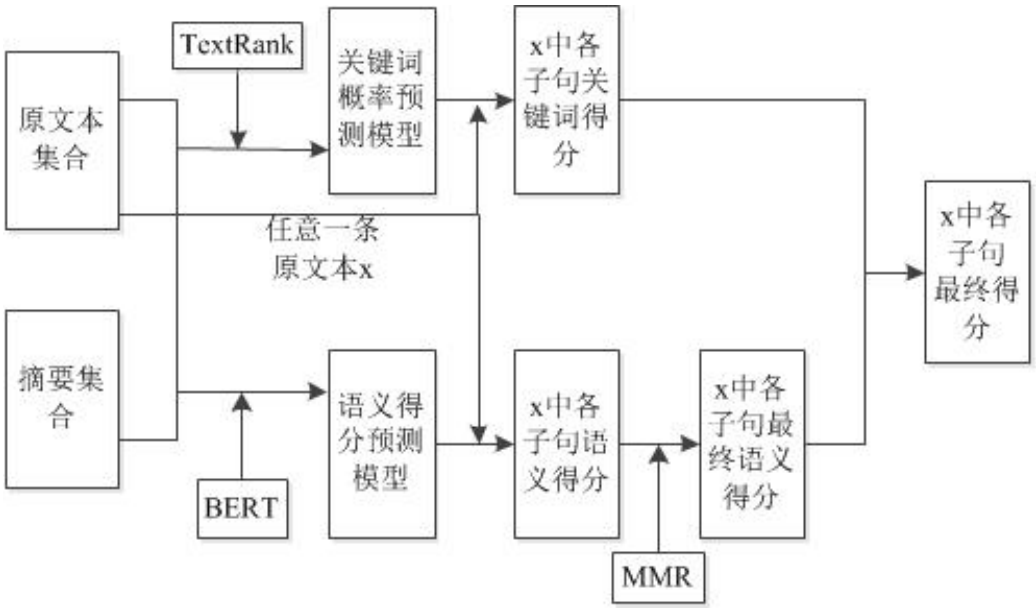

图 3-9 第一阶段抽取文本子句过程

由图 3-10 可以看出，此方法一方面从子句的语义方面评估其重要性，一方面从子句的关键词方面评估其重要性。在子句的语义得分方面，首先使用 BERT 获得原文本中每个子句的语义向量及标准摘要的向量，以此计算得到每个子句与标准摘要的语义相似性，然后基于此相似性对原文打标签得到数据集 T_S，之后使用 T_S 按照序列标注的思想训练得到一个语义得分预测模型，模型训练好之后即可用它获得文本摘要数据集中每条原文本的各子句语义得分。此外，考虑到挑选出的子句的多样性，又使用 MMR 算法对上述语义得分进行调整，获得文本中子句最终的语义得分 $score\_sim = \{s\_s_1, s\_s_2, \ldots, s\_s_{T_S}\}$，其中 s_s_k 表示使用 MMR 算法调整之后第 k 个子句的语义得分，T_s 是原文本的子句数；在子句的关键词得分方面，首先使用 3.3 节中的方法获得原文中每个词语属于关键词的概率，然后基于此概率通过计算获得文本中所有子句的关键词得分 $score\_key = \{s\_k_1, s\_k_2, \ldots, s\_k_{T_S}\}$，其中 s_k_k 表示第 k 条子句的关键词得分；最终同时考虑子句的语义得分 $score\_sim$ 和关键词得分 $score\_key$ 获得原文本中每个子句最终的得分 $score = \{s\_c_1, s\_c_2, \ldots, s\_c_{T_S}\}$，然后根据此得分选取出得分最高的 K 个子句。具体每步的做法如下：

a) 获得子句语义得分

首先是文本子句语义向量和文本主题向量的获取。

对于子句语义向量的获取，此处的做法与 3.4 节中的方法一致。即在文本子句间添加特殊符号“CLS”，然后使用 BERT 模型输出层中此特殊符号所在位置的输出值作为各子句的语义向量，文本子句向量序列表示为 $s\_v = \{s\_v_1, s\_v_2, \ldots, s\_v_{T_S}\}$。

对于文本主题向量的获取，一般都是对文本中的词向量或者子句向量进行相关操作得到，但是，一方面，简单使用文本中的词向量或子句向量进行拼接或者进行相关计算获得的文本主题向量不准确，另一方面，文本中有些内容与文本的主题是没有关联的，所以并不需要将其所携带的信息编码到文本主题向量中。所以，本文并不使用原文，而是使用原文本所对应的标准摘要来获得文本主题向量。具体地，对于标准摘要而言，将其按字符切分，但不对其进行切分子句，而是将其作为一个整体，然后使用 BERT 获得其语义向量，作为文本的主题向量。

获得子句和标准摘要的语义向量后，接下来是制作数据集 T_S，训练语义得分预测模型。首先使用余弦相似度计算每条子句与标准摘要，即与文本主题的相似性，根据相似性选取出相似性最高的 K_{ss} 条子句并得到原文本的一个标签序列 $s\_t=\{s\_t_1, s\_t_2,\ldots, s\_t_{Ts}\}$，其中 $s\_t_k\in\{0,1\}$，表示第 k 条子句的标签，若它是上述被挑选出的 K_{ss} 个句子之一，则 $s\_t_k=1$，否则，$s\_t_k=0$。以此方法得到数据集 T_S，T_S 中每条数据为(s_v, s_t)。然后使用此数据集按照序列标注的思想训练得到语义得分预测模型。

模型训练完成后，对于任意一条原文本而言，先使用 BERT 获得其子句向量序列，将其输入到模型中，根据模型输出层的结果即可得到文本中每条子句属于标签“1”的概率，即该条子句与文本主题的语义相似性。

最后考虑到挑选内容的多样性，使用最大边界相关算法（Maximal Marginal Relevance,MMR）算法对子句的语义得分进行调整，得到文本每条子句的最终语义得分。

MMR 是一种推荐重排算法，其思想是在保证排序结果相关性的同时减少排序结果的冗余性。MMR 算法使用式 3-26 的方式使排序结果的相关性和多样性达到平衡。

$$\mathrm{MMR}(\mathrm{Q},\mathrm{S},\mathrm{R})=\mathrm{Arg}\max_{s_i\in S\setminus R}\left\{\lambda sim_1(s_i,Q)-(1-\lambda)\max_{s_j\in R}sim_2(s_i,s_j)\right\} \quad (3\text{-}26)$$

式中 Q 在本任务中相当于文本的主题含义，即标准摘要的内容；S 表示候选集合，R 表示已得到的以相关度为基础的候选集合，s_i 表示候选集合中的第 i 个元素，在本任务中相当于文本中的每条子句；$sim_1(s_i,Q)$表示候选集合中的元素与查询内容 Q 的相似度，即 s_i 的相关性；$sim_2(s_i, s_j)$表示候选集合中的元素与已选集合中某个元素的相似度，即 s_i 的冗余性。在本任务中，$sim_1(s_i,Q)$即使用序列标注模型预测得到的每个子句与文本主题的相似性，$sim_2(s_i, s_j)$是根据子句 s_i、s_j 的句向量使用余弦相似度算法计算得到的两者的相似性。

b) 获得子句关键词得分

传统抽取式摘要做法中对于子句关键词得分都是基于子句中含有的关键词数量进行统计得到的，但是对于同属文本关键词集合中的词语而言，不同词语的

关键程度并不一样，所以使用关键词含量获得子句重要性并不准确。此外，对于生成式摘要而言，由于数据集中的每条原文都对应有标准摘要，且标准摘要中的词语一定是文本的关键词，则可以使用标准摘要中的词语对关键词提取算法做指导。所以本文最终使用与 3.3 节中同样的方法获得原文本中每个词语属于关键词的概率。

获得原文中每个词语属于关键词的概率之后，将原文按照“!。？”进行切分子句，然后对于每个子句而言，对其中的词语按照它属于关键词的概率进行降序排序，最后选择概率最高的 K_{sk} 个词语的概率和作为此子句的关键词得分。对于 K_{sk}，本文统计了每条原文本关键词的概率值情况、每个子句中含有其原文本关键词数量的情况及每个子句中关键词占比的情况，最终决定使用关键词占比决定 K_{sk} 的值，即 K_{sk}=0.3*子句中的词语数。

对于一条原文本 x，获得其各子句关键词得分的具体步骤如表 3-3 所示。

表 3-3 子句关键词得分步骤

原文： $x=\{x_1, x_2, \ldots, x_{Tx}\}$		
步骤	操作	结果
1	使用 Word2vec 获得词向量	$x\_v = \{x\_v_1, x\_v_2, \ldots, x\_v_{Tx}\}$； x_v_i 代表第 i 个词语的词向量
2	使用 3.3 节中的关键词概率预测模型获得文本中每个词语属于关键词的概率	$x\_p = \{x\_p_1, x\_p_2, \ldots, x\_p_{Tx}\}$； x_p_i 代表第 i 个词语属于关键词的概率
3	对原文切分子句	$s = \{s_1, s_2, \ldots, s_{Ts}\}$； s_k 表示原文中第 k 个子句，T_s 表示原文子句数
4	对每个子句分词	$s_k = \{x^k_1, x^k_2, \ldots, x^k_{Tsk}\}$； T_{sk} 表示第 k 个子句中的词语数
5	对每个子句按照词语属于关键词的概率进行降序排序	$p_{sk} = \{p^k_1, p^k_2, \ldots, p^k_{Tsk}\}$ p^k_i 表示第 k 个子句中关键词概率排名第 i 的词语的关键词概率值
6	获得每个子句的关键词得分	$s\_k_k = \sum_{i=1}^{K_{sk}} p_i^k$ $K_{sk} = 0.3 * T_{sk}$

c) 获得每个子句最终得分

获得原文中每个子句的语义得分 *score_sim* 和关键词得分 *score_key* 后，为了同时从多个方面考虑子句的重要性，本文对子句的两种得分进行了权衡。首先对两个得分进行了归一化，然后对其分配不同的权重，获得每个子句最终的得分

score。此部分做法可根据式 3-26 至 3-28 进行理解。

$$score\_sim' = norm(score\_sim) \quad (3\text{-}26)$$

$$score\_key' = norm(score\_key) \quad (3\text{-}27)$$

$$score = \alpha * score_{sim}' + (1-\alpha) * score\_key' \quad (3\text{-}28)$$

式 3-26 和 3-27 中的 *norm(.)*表示归一化操作，此处使用最大最小归一化将 *score_sim* 和 *score_key* 中的元素值调整到 0 到 1 之间。式 3-28 中 $\alpha \in (0,1)$，α 越大，表示在抽取子句的时候，更偏向于考虑子句与文本语义的相似性。

3.6 模型实现

本文实现的带有注意力机制和指针网络的 Seq2Seq 模型在进行摘要生成时主要包括 4 个实现步骤，具体如图 3-10 所示。

1) 构建嵌入层，将输入文本数据化；
2) 构建编码器，对输入进行编码，获得隐藏层最后一层各时刻的输出值 *enc_outputs* 以及隐藏层中每层网络最后一个时刻的状态值 *enc_state* 作为编码器的输出；
3) 构建解码器，根据 *enc_outputs*、*enc_state* 等，调用 attn_decode()获得每一解码时刻的注意力分布 *attn*，然后获得上下文语义向量 *c*，进而获得生成概率 p_{gen}、预测得到的模型词汇表中词语的概率分布 p_{gv} 等；
4) 根据 *attn* 获得复制概率 p_{lv}，然后调用 cal_final_dist()函数根据 p_{gen}、p_{lv}、p_{gv} 计算模型最终的预测概率 *p*。

图 3-10 摘要生成步骤

本文所提出的两种生成模型的创新点主要体现在步骤 3)中 attn_decode()函数的定义与实现中，所以接下来本文将对两种模型中此函数的实现进行介绍。

首先是基于关键词改进注意力机制的文本摘要模型，此时 attn_decode()函数的定义为 attn_decode1(*decoder_state,enc_outputs,key_probs,decoder_input*)。其中 *decoder_state* 表示解码器此时刻的隐藏状态，在解码的第 0 时刻，使用编码器藏隐层最后一个时刻的状态值 *enc_state* 进行初始化。*enc_outputs* 表示编码器隐藏层最后一层各时刻的输出值。*key_probs* 表示使用本文设计的关键词概率预测模型预测得到的原文本中每个词语属于关键词的概率。*decoder_input* 表示解码器上一时刻预测得到的结果。

attn_decode1()函数的内部计算过程见图 3-11 所示。

1) *decoder_state* 与 *enc_outputs* 进行相关计算得到注意力分布 *_attn*；
2) 使用 *key_probs* 调整 *_attn* 得到调整后的注意力分布 *attn*；
3) *enc_outputs* 与 *attn* 加权求和得到环境向量 c_t；
4) c_t、*decoder_state* 和 *decoder_input* 进行操作 1 得到生成概率 p_{gen}，c_t、*decoder_state* 和 *decoder_input* 进行操作 2 获得模型在系统词汇表上的预测概率分布 p_{gv}。

图 3-11 attn_decode1 函数运行步骤

对于双编码器文本摘要模型，由于使用了两个编码器，则需要在图 3-10 中的步骤 2)后添加一个构建子句级别编码器的过程，其构建方法与输出结果与步骤 2)类似。此时 attn_decode() 函数的定义为 attn_decode2(*decoder_state*,*enc_outputs*,*subsents_enc_outputs*, *decoder_input*)。其中，参数名与 attn_decode1()函数中相同的参数的含义一致，*enc_outputs* 表示词级别编码器的输出，*subsents_enc_outputs* 表示子句级别编码器的输出。

attn_decode2()函数内部计算过程见图 3-12 所示。

1) *decoder_state* 分别与 *enc_outputs*、*enc_outputs* 进行计算得到词注意力分布 *attn_word* 和子句注意力分布 *attn_subsent*；
2) *enc_outputs* 与 *attn_word* 加权求和得到词级别上下文语义向量 *ct_w*，*vec_enc_outputs* 与 *attn_subsent* 加权求和得到子句级别上下文语义向量 *ct_s*；
3) *ct_w*、*ct_s* 与网络中相关参数计算得到门值 *gate*，然后根据 *ct_w*、*ct_s* 及 *gate* 得到环境向量 c_t；
4) c_t、*decoder_state* 和 *decoder_input* 进行操作 1 得到生成概率 p_{gen}，c_t、*decoder_state* 和 *decoder_input* 进行操作 2 获得模型在系统词汇表上的预测概率分布 p_{gv}。

图 3-12 attn_decode2 函数运行步骤

此时，当解码第 0 时刻时，由于模型中有两个编码器，所以需要将两个解码器最后时刻的状态值 *enc_state*、*subsent_enc_state* 使用 reduce_state()函数进行合并，然后再对 *decoder_state* 进行初始化。合并的方法是将两者拼接然后使用一个全连接层进行降维。

3.7 本章小结

本章首先基于数据集文本中所存在的问题，介绍了对数据进行处理的方法，然后对本文所提出并实现的几种文本摘要模型进行了详细介绍，包括基于关键词改进注意力机制的生成式文本摘要模型、拥有两个编码器的生成式文本摘要模型

以及基于子句语义得分和关键词得分提取文本关键内容的双阶段式文本摘要模型，最后对模型实现中所涉及的重要方法进行了说明。

第四章 文本分类技术研究

本章主要针对舆情数据中信息杂乱、无法利用其中有价值信息的问题研究了文本分类技术，并具体针对成都地区的警情信息进行了数据获取和分析，然后提出了一种堆叠式的文本分类模型 SPCNN（Subsequence Proposed Convolutional Neural Network,SPCNN）。该模型的每一层主要由特征提取子网络（Feature Extraction Subnetwork,FESN）、分类子网络（Classification Subnetwork,CSN）和子序列建议子网络（Subsequence Proposed Subnetwork,SPSN）组成，FESN 负责提取特征，CSN 负责分类，SPSN 负责从上一层网络的输入中截取更具有表征性的子序列供下层网络输入。

4.1 数据集的获取和数据预处理

本文首先根据先验知识制定了警情数据的类别以及每类数据的触发词，具体如表 4-1 所示。然后从媒体、微信、新浪微博等平台上使用爬虫技术根据这些触发词获得了大量的网络数据。

表 4-1 各警情类别触发词

类别	触发词集合
偷盗类	盗窃、偷盗、偷窃、扒窃、摸包包、手机
两枪类	劫、抢、夺、
人身伤害类	杀、伤、遇害、被害、行凶、坠楼、跳楼、溺水、溺亡、打架、群殴、殴打、绑架、斗殴、猥亵、强暴、投毒
黄赌毒类	毒品、冰毒、海洛因、贩毒、笑气、罂粟、吗啡、K 粉、赌、赌博、老虎机、投注、博彩、嫖、嫖宿、嫖娼、卖淫、扫黄
诈骗类	诈骗、行骗、骗钱、被骗、骗局、骗、假钞、假币、跑路、勒索、跑路
传销类	组织、传销、窝点、
毁坏财物类	烧毁、砸毁、打砸、破坏、焚毁、破坏、砸、
交通事故类	追尾、事故、肇事、撞车、翻车、醉驾、酒驾、车祸、侧翻
安全事故类	火灾、爆炸、燃烧、中毒、剧毒、自燃
扰乱公共秩序和危害公共安全	公共安全、聚众闹事、妨碍执法、黄牛、倒票、票贩子、公共秩序、非法飞行、黑飞、飙车、扰乱公共秩序、非法出版
维权类	投诉、施工、装修、房、权益、维护

由于本文最终的目的是对成都地区的警情信息进行分类，所以本文首先对爬取到的数据进行了地点过滤处理。然后针对部分数据中触发词周围存在假设性词语的情况，如“…，就有可能导致侧翻、被追尾等交通事故，严重的还有可能造成重大交通事故”，虽然文本内容中包含“侧翻”和“追尾”这两个触发词，但这些触发词之前的“可能”表明这一情况实际上并未发生，所以本文对未触发数据进行了过滤。随后针对文本中存在警情触发词，但实际上并不属于警情数据的问题，对数据进行了非警情数据过滤，如将交通信息播报类数据和消防演练类数据剔除掉。之后又对警情数据中存在大量对于同一事件重复报道的问题对数据进行了去重处理。数据处理的具体操作如表 4-2 所示。

表 4-2 数据过滤处理

步骤	处理	使用技术
1	地点过滤	使用 CRF++工具提取出文本中的地点词，然后将非成都地区的数据过滤掉；
2	未触发数据过滤	将与“可能”、“好像”等假设性词语在同一个子句中，且出现在假设性词语之后的触发词进行遮掩，即将其替换为一个统一的符号“###”，然后再检查此条文本中是否仍含有触发词，若没有则将此条文本过滤掉；
3	非警情数据过滤	使用 TextCNN 模型进行二分类，将非警情数据过滤掉；
4	去重处理	参照文献[70]中的做法，先使用事件信息进行简单对比过滤，再使用 VSM 类模型从语义级别对比将重复数据剔除。

数据集获取后，对数据集中的每条数据进行分词、去停用词，然后按字进行统计，取频数位于前 *vocab_size* 的字构成字汇表，并为字汇表中的每个字随机初始化一个字向量。然后将数据集按比例划分为训练集、验证集和测试集，并对每个数据集中的数据进行如下处理：首先对数据集进行批处理，形成多个 *Batch*，每个 *Batch* 中的数据为 *batch_size* 条；由于神经网络模型每次以一个 *Batch* 的数据为单位进行学习，所以接下来需要将每个 *Batch* 中的数据在字符级别处理成长度一致；最后将每个 *Batch* 中的每条数据从自然语言文本转化为数据矩阵的形式，矩阵中的每一行是文本中一个字的字向量，矩阵的行数是文本的长度，矩阵的列数与字向量的维度一致。每个数据集中数据处理的具体操作如表 4-3 所示。

表 4-3 各数据集中数据处理操作

步骤	处理	具体操作
1	分词、去停用词	使用 jieba 工具进行分词，使用哈尔滨工业大学公开的停用词表进行去停用词。然后对文本按字切分，构建模型字汇表；
2	生成字向量	使用随机初始化为字汇表中的每个字生成一个字向量；
2	批处理	将数据化分成若干个 *Batch*，每个 *Batch* 的大小为 *batch_size*；
3	每个 *Batch* 的数据处理成长度一致	按字符统计出每个 *Batch* 中最长数据的长度，然后对于比此长度长的数据，从文末进行截取，比此长度短的数据，在文末补相应个数的‘<PAD>’符号；将‘<PAD>’符号添加到字汇表中并为其初始化一个向量；
4	文本转化为数据矩阵	对于一个 *Batch* 中的每条数据，从系统字汇表中取出文本中每个字相对应的向量按顺序进行拼接，形成数据矩阵。

4.2 文本分类模型 SPCNN

4.2.1 模型的提出

在文本分类技术中，大多数模型会将文本的全部内容作为网络的输入，以对全文的语义进行相应的理解和学习，但是当文本内容过长或者分类的粒度较精细化的时候，使用文本的局部内容就可以预测得到文本的所属类别，而且由于输入的文本更短、信息更加集中，分类的效果也会更优。

例如，当对舆情数据从“城市形象、交通相关、旅游相关、社会治安、……”等大的类别进行分类时，由于这些数据存在局部相似的情况，所以需要从全文对文本内容进行理解才可以明确其类别。如表 4-4 中的 1、2 条数据，当读到“红绿灯设计不合理”以及“长期出于拥堵情况”时，并不能判断数据的所属类别，因为它们都有可能属于“交通相关”类，但是当读到“你的言论影响四川形象”及“希望能够早日实现便民出行”时，便可以确定其所属类别了。再如表中第 3 条数据，“无论怎么搞旅游业”使得数据看起来属于旅游相关类，但根据文本末尾的“城市建设……”内容可以知道，其实此条数据应该属于城市形象类。

表 4-4 粗粒度文本分类举例

类别	文本
城市形象	经常看到很多朋友都在抱怨说，发现成都很多地方的红绿灯设计不合理。有的路段红灯等的打瞌睡，才能换来几秒绿灯；有的绿灯时长太短了，人得用百米冲刺的速度才能走完；有的红绿灯倒计时还没结束就突然跳到红灯了，刹车都来不急；有的灯晚上都没人但是还是要停下来等很久……在成都的大街小巷有没有哪个红绿灯让你很无奈，或者存在着交通隐患的，欢迎各位麻友跟帖指出来！你的言论影响四川形象。
交通相关	尊敬的领导您好惠王陵地铁站出站口的一条干道，只有 2 车道，常期处于拥堵情况，更不要说是在上班出行高峰期；地铁 A 口本就要拐弯，两车道不足以支持频繁出现的公交车及货车之间的会车，一会车就是拥堵的产生，且外东洪路与驿都大道的交叉路口的红绿灯时间较短，在高峰时段根本无法缓解交通压力。希望能够早日实现便民出行，与街对面 468 的发展所匹配，大家一起高速发展
城市形象	成都政府缺少底蕴，决定了成都市无论怎么搞旅游业，无论怎么网红扩充也成为不了一个一线城市。房地产问题始终无法解决，纵容地产商肆意妄为，天天到处维权并起，胡乱制定政策，朝令夕改，推诿，踢皮球。城市建设除了闹天府新区的噱头，其他地方经期性挖休。可能全国都这样吧。

但是当对此数据中的社会治安类按照表 4-1 中所示的类别再进行细分类时，使用局部信息便可得到其所属类别。如表 4-5 中所示，分别根据 1-3 条数据中的“我爸爸下去他就开始打我爸，拉他，然后用砖头打我爸”、“今天晚上六点左右，女朋友在盐市口被骗了”、“我新买的电瓶车被盗，位置在成都高新区东苑附近”这些文本中的子序列内容就可以判断其分别属于人生伤害类、诈骗类和偷盗类。

表 4-5 细粒度文本分类举例

类别	内容
人身伤害类	你好，我是崇州市三江镇听江村 10 组的王蝶，我爸爸叫王术文。今天下午我们附近有个理疗店的老板，姓鲁，在地邻上我叫他哥哥，他叫我爸舅舅，我爸爸在家开了家理发店，今天不舒服关了门在楼上睡觉，他前两天把车停在我爸爸铺子门口，开回去了发现他的车被人划了，今天来找我爸，说是我爸爸划的，从楼下就开始骂，叫我爸下去，然后损坏了我爸铺子上的理发用具，我爸爸下去他就开始打我爸，拉他，然后用砖头打我爸，我在拉他们和阻挡鲁的时候眼镜被打烂了，把脸划了。今天报了警，崇州的警察过来拍了照，了解了一下，让我们自行协商……
诈骗类	#成都爆料#一个自称香港的人，出差来成都，身上没有人民币，找女朋友兑换人民币，通过网银转账方式，说是网银转账要延时二个小时到账，给了网银凭证给我女朋友看，我女朋友稀里糊涂的就相信了，被教育了 1000 人民币！！！过程我也不是特别清楚，反正就是今天晚上六点左右，女朋友在盐市口被骗了，希望各位提高警惕，如果遇到类似的请告诉我，我想教育教育这个坑”
偷盗类	2018 年 5 月 22 日我新买的电瓶车被盗，位置在成都高新区东苑附近！报警到现在已经一周了，还没有任何消息！当天被盗的还有两个摩托车！派出所碰到的，几乎在差不多时间被盗！成都偷车贼这么猖狂，警察几乎不办事，立个案就没讯息了！监控也是第三天才去看的……

基于以上分析，文本提出了一种新颖的文本分类模型 SPCNN。SPCNN 是一种堆叠模型，即整个模型由两层网络堆叠构成，每层网络的输入是同一条文本的不同子序列。模型每层网络的结构都相同，但是每层网络中的参数并不共享，因为不同的网络要适应不同的输入内容进行学习。图 4-1 展示了 SPCNN 模型结构。

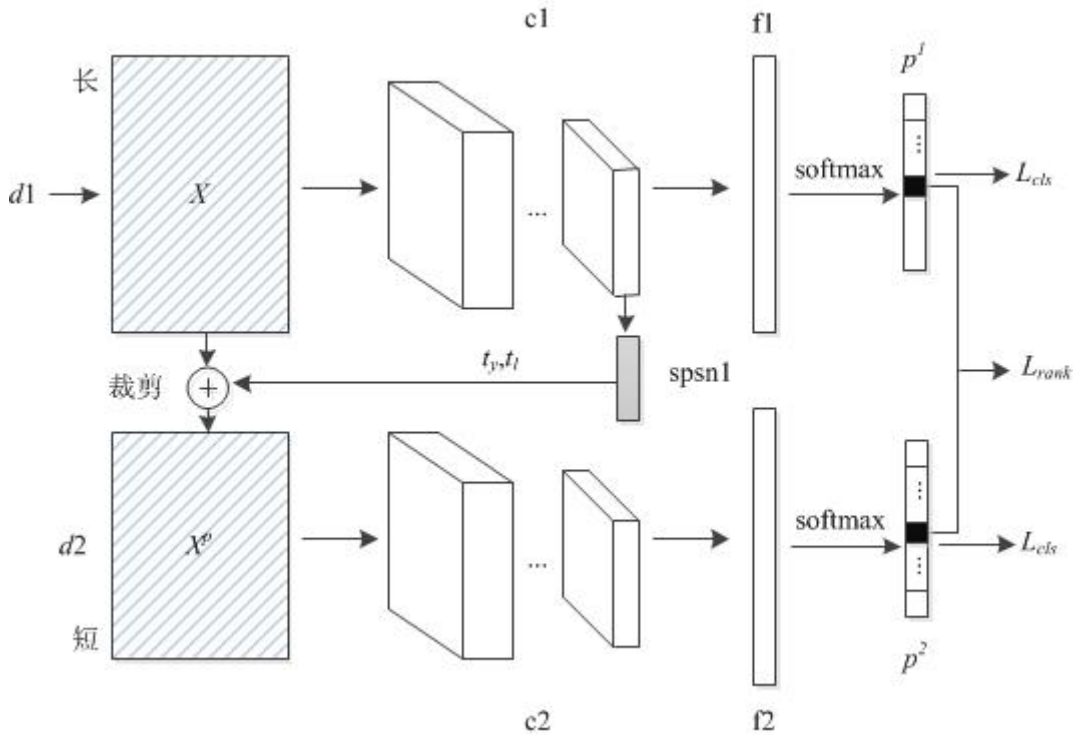

图 4-1 SPCNN 模型结构

由上图可知，模型的每层网络包括三个部分，一个是特征提取子网络 FESN（即图 4-1 中的 c1、c2），一个是分类子网络 CSN（即图 4-1 中的 f1、f2 及对应的 Softmax 层），一个是子序列建议子网络 SPSN（即图 4-1 中的 spsn1），其中每层网络的特征提取子网络及分类子网络构成了一个 TextCNN 网络。分类子网络保证每层网络都有一定的类别预测能力，子序列建议子网络可以保证为下一层网络从上层网络的输入中截取出一个更具有表征性的子序列作为输入。更深层的网络的输入更短、信息更集中，则网络提取到的文本特征也更准确，分类的效果也更优。

模型第一层网络的输入是 $d1$，第二层网络的输入 $d2$ 是 $d1$ 的子序列，它是从 $d1$ 中截取出的对于分类而言更具有表征性的内容。X 表示 $d1$ 经过模型嵌入层后的结果。由上图可知，数据首先经过特征提取子网络以提取出相关的文本特征，然后，提取出的特征一方面经过 CSN 网络预测得到此条数据在所有类别上的概率分布。另一方面，经过 SPSN 网络建议一个子序列，然后从 X 中获得相应的数据作为下层网络的输入。

由上述可知，整个模型每一层网络主要包括两个任务，一个是对数据进行分类，一个是从原输入序列中建议一个子序列。两个任务的第一步操作是相同的，即对于 X，首先经过特征提取子网络得到其抽象特征，此操作可以表示为 $W\&X$，其中 $\&$ 表示卷积、池化及激活操作，W 表示上述操作中涉及到的所有网络参数。接下来本文将对这两个任务及模型的学习过程进行详细介绍。

4.2.2 模型的学习与训练

模型的第一个任务由每层网络的 CSN 网络来完成，它主要是根据特征提取子网络提取出的特征预测 X 在所有类别上的概率分布 p。所以，第一个任务的操作过程可以表示为式 4-1：

$$p(\mathrm{X}) = \mathrm{b}(W \& X) \tag{4-1}$$

其中 b(.)表示全连接层和 Softmax 层，全连接层的作用是将特征提取子网络得到的特征向量映射为一个维度与分类类别数相同的向量，Softmax 层的作用是将此向量转化为一个概率分布。

网络的第二个学习任务由 SPSN 网络完成，它主要根据特征提取子网络提取出的特征从原序列 $d1$ 中建议一个子序列，即 $d2$，供下层网络学习。此部分过程如式 4-2 所示。

$$t_y, t_l = o(W\&X) * len(d1) \tag{4-2}$$

式中 *o(.)*表示非线性操作，其输出是一个二维向量。*len(.)*表示求长度操作，此处即得到 $d1$ 的字符数。t_y 表示 SPSN 网络建议的子序列的中心位置，t_l 表示此中心位置与此子序列起始位置的距离。为了避免 SPSN 网络建议的子序列过短，导致其包含的内容过少，网络不能提取出充分的文本信息，所以本文约束 t_l 与 *len(d1)*的比例不能小于 1/3。得到 t_y 和 t_l 后，则可以得到子序列的起始位置 t_s、t_e，其计算方式如式 4-3 和 4-4 所示。

$$t_s = t_y - t_l \tag{4-3}$$

$$t_e = t_y + t_l \tag{4-4}$$

得到子序列的起始位置后，本文使用一种掩盖操作来将 $d2$ 所对应的数据从 X 中裁剪出来。裁剪操作可以表示为式 4-5。

$$X^p = X \odot \mathrm{M}(t_y, t_l) \tag{4-5}$$

式 4-5 中 X^p 表示裁剪后的数据，⊙表示逐元素相乘，*M(.)*表示掩盖操作，这种掩盖操作既可以在前向传播过程中将原序列中具有表征性的内容截取出来，又可以保证 SPSN 网络在反向传播的过程中能够被优化。*M(.)*通过使 $d2$ 中每个字符在 X 中所对应的向量保持不变而将 X 中其他向量置为零向量实现对 X 的裁剪，其具体公式如式 4-6 所示。

$$M(.) = h(y - t_s) - h(y - t_e) \tag{4-6}$$

式中 y 对应 $d1$ 中每个字符的位置索引，式 4-7 表示了 *h(.)*的计算过程，当 *h(.)*中 k 的取值足够大时，*h(.)*逼近为一个阶跃函数。

$$h(z) = 1/\{1 + \exp^{-kz}\} \tag{4-7}$$

模型最终被上述两个任务所对应的两个有监督学习所共同优化，交替的学习数据中更具有区分度的特征以进行分类以及建议更具有表征性的子序列以供下

层网络使用。两个监督学习的损失函数分别定义为层次分类损失和层间排序损失，模型最终的损失函数如式 4-9 所示。

$$\mathrm{L(X)} = \sum_{i=1}^{scale}\{L_{cls}(p^i, \mathrm{Y^*})\} + \sum_{i=1}^{scale-1}\{L_{rank}(p_t^i, p_t^{i+1})\} \quad (4\text{-}9)$$

式中 *i* 表示第 *i* 层网络，*scale* 表示模型总共的网络层数，在文本中即 2。Y* 代表数据在所有类别上的真实概率分布，是一个 one-hot 向量，p^i表示模型第 *i* 层网络预测得到的数据在所有类别上的概率分布。L_{cls} 代表层次分类损失，主要用于优化图 4-1 中 c1、c2、f1 和 f2 结构中涉及到的参数，确保每层网络具有足够的类别预测能力，其计算方式是在 p^i 和 Y*之间使用神经网络中常用的 Softmax 损失函数。L_{rank} 表示层间排序损失，p^i_t 表示第 *i* 层网络在正确类别 *t* 上预测得到的概率值。排序损失主要用于优化图 4-1 中 spsn1 结构中涉及到的参数，其具体计算公式如式 4-10 所示。

$$L_{rank}\left(p_t^i, p_t^{i+1}\right) = \max\{0, p_t^i - p_t^{i+1} + margin\} \quad (4\text{-}10)$$

此损失函数使得在训练的过程中有 p^{i+1}_t - p^i_t>*margin*，即使用第 *i* 层网络的预测结果作为参考，通过强制第 *i*+1 层网络预测结果的置信度比第 *i* 层网络预测结果的置信度更高来使得 SPSN 网络建议的子序列是原序列中更加具有表征性的内容。

最终，根据图 4-1 将模型搭建好后，便可以对模型进行训练。模型的训练过程包括以下三个步骤：

1) 对 FESN 网络进行预训练。首先搭建一个 TextCNN 模型并使用分类数据集对此模型进行训练，模型训练完成后，将模型中全连接层之前的网络所对应的网络参数加载到本文模型的对应网络结构中，即图 4-1 中的 c1、c2。
2) 对 SPSN 网络进行预训练。用第一层网络中 FESN 网络提取出的特征中的最高响应值来确定第一层网络输入中更具有表征性的子序列，然后以此作为参考子序列训练 SPSN 网络以对图 4-1 中 spsn1 的参数进行初始化。
3) 交替的使用上文所述的两种监督学习对模型中的参数进行优化。首先，保持 SPSN 网络的参数不变，训练 FESN 网络和 CSN 网络，即优化式 4-9 中第一项 *scale* 个损失函数 L_{cls}，使其达到收敛；第二步，固定 FESN 网络和 CSN 网络参数保持不变，训练 SPSN 网络，即优化式 4-9 中第二项 *scale*-1 个排序损失函数 L_{rank}，直至其收敛。这两个学习过程迭代进行，直到损失函数 *L(X)*中的两部分都达到收敛为止。

一旦 SPCNN 模型训练完成，则可以使用不同层次的网络从原文的不同子序列中抽取得到多个原文特征向量，其可以被表示为一个如式 4-11 所示的集合：

$$\{F_1, F_2, \ldots, F_{scale}\} \quad (4\text{-}11)$$

其中 F_i 表示第 *i* 层网络的 FESN 网络所提取到的文本特征。为了利用特征融

合的优点，本文将不同层网络提取出的特征进行了融合以获得更丰富的文本特征，然后使用 Softmax 回归进行分类。文本中特征融合的方式是将不同特征向量进行拼接。

4.3 模型实现

此部分主要对本文中提出的 SPCNN 模型中的关键实现过程进行介绍。

SPCNN 模型的构建方法如图 4-2 所示。

```
class RACNN(nn.Module):
    def __init__(self, vocab_size ,embedding_dim, filter_sizes, filter_num, num_classes,
                 pretrained):
        super(RACNN, self).__init__()
        self.embedding = nn.Embedding(vocab_size + 1, embedding_dim)
        self.c1 = text_bn(embedding_dim, filter_sizes, filter_num, pretrained)
        self.c2 = text_bn(embedding_dim, filter_sizes, filter_num, pretrained)
        self.spsn1 = nn.Sequential(
            nn.Linear(len(filter_sizes)*filter_num, 256),
            nn.Tanh(),
            nn.Linear(256, 2),
            nn.Sigmoid(),
        )
        self.classifier1 = nn.Linear(len(filter_sizes) * filter_num, num_classes)
        self.classifier2 = nn.Linear(len(filter_sizes) * filter_num, num_classes)
        self.classifier12 = nn.Linear(2 * len(filter_sizes) * filter_num, num_classes)
        self.dropout = nn.Dropout(0.5)
```

图 4-2 SPCNN 网络的构建

首先是构建模型的嵌入层 self.embedding，参数 *vocab_size* 代表模型字汇表的大小，*embedding_dimension* 表示嵌入层字向量的维度。

然后是调用 *text_bn()*函数构建模型中的 FESN 网络 self.c1、self.c2，即图 4-1 中的 c1、c2。*filter_sizes* 表示卷积核窗口的大小。*filter_num* 表示卷积核的数目。*pretraied* 用以决定是否使用预训练的 TextCNN 模型中的参数对 c1、c2 中的参数进行初始化，当 *pretraied* 为一个已经训练完成的 TextCNN 模型的存储路径时，便将此模型中的参数加载出并将其中卷积层参数值加载到 c1 和 c2 中；当 *pretrained*=False 时，便对 c1 和 c2 中的参数进行随机初始化。*num_classes* 代表

分类的类别数。

随后是构建 SPSN 子网络 self.spsn1，由上述实现过程可知，本文中使用两个全连接层实现 SPSN 网络，网络的输出为两个变量，即式 4-2 中 *o(W&X)*的结果。

最后是构建每层网络的 CSN 网络 self.classifier1 和 self.classifier2，即图 4-1 中的 f1 和 f2。self.classifier12 代表使用融合特征进行分类的分类器。

SPCNN 模型的训练过程包括分类训练过程和 SPSN 网络训练过程，分类训练过程包括优化图 4-1 中的 c1、c2、f1 和 f2 中的网络参数，SPSN 网络训练过程优化图 4-1 中的 spsn1 中的网络参数。SPCNN 模型的训练过程如图 4-3 所示。

初始化：

- 判断分类损失或者排序损失达到收敛的阈值$\theta 1$
- 分类训练过程、SPSN 网络训练过程中分别需要优化的网络参数集合 cls_params、spsn_params 及优化所使用的优化器 opt1、opt2；
- 分类训练、SPSN 网络训练上一轮迭代损失函数的结果 cls_old_loss、spsn_old_loss，分类训练、SPSN 网络训练本轮迭代损失函数的结果 cls_new_loss、spsn_new_loss
- SPCNN 模型的训练迭代轮次 iteration 及最大迭代次数$\theta 2$
- 分类训练、SPSN 网络训练各自的迭代轮数 cls_iter 和 spsn_iter 及最大迭代轮数$\theta 3$

训练过程：

1) 若 old_cls_loss - new_cls_loss>阈值$\theta 1$ 且 old_spsn_loss - new_spsn_loss>阈值$\theta 1$ 且 iteration<阈值$\theta 2$，则转至步骤 2)，否则模型训练结束；
2) 若分类训练过程的训练轮次 cls_iter 大于$\theta 3$，则将 cls_iter 置零，转步骤 4)。否则，转至步骤 3)；
3) 将 new_cls_loss 赋值给 old_cls_loss，进行分类训练。根据每层网络的分类结果使用 multitask_loss()计算得到分类损失 new_cls_loss，然后进行反向调节，使用 opt1 对 cls_params 集合中的参数进行更新。cls_iter 加 1，iteration 加 1，转至步骤 2)；
4) 若 SPSN 网络的训练轮次 spsn_iter 大于$\theta 3$，则将 spsn_iter 置零，转步骤 1)。否则，转至步骤 5)；
5) 将 new_spsn_loss 赋值给 old_spsn_loss，进行 SPSN 网络的训练。根据每层网络的分类结果使用 pairwise_ranking_loss()计算得到排序损失 new_spsn_loss，然后进行反向调节，使用 opt2 对 spsn_params 进行更新。spsn_iter 加 1，iteration 加 1，转至步骤 4；

图 4-3 SPCNN 训练过程

4.4 本章小结

本章首先对本文所使用的分类数据集以及此数据集的获取方式进行了介绍，并说明了对数据进行的预处理操作，然后对文本模型提出的动机以及模型结构进行了介绍，最后对模型学习过程以及模型训练方式进行了详细的介绍。

第五章 相关实验测试

5.1 实验背景

本章分别对第三章所提出的几个文本摘要模型以及第四章所提出的一个文本分类模型进行了相关效果评估。在文本摘要模块，针对生成式模型，本文使用了 4 种模型与本文所提出的两种模型进行了对比和分析。针对双阶段式模型，本文首先测试了模型中各个参数的设置对抽取出内容的影响，然后设计并测试了其他三种双阶段式模型进行对比实验。在文本分类模块，本文首先分析了模型参数设置以及预训练操作对模型的影响，然后设计了对比试验进行效果评估。各实验所使用的数据集及评估方法将在其各自小节中具体介绍。

本文中各实验的实验环境如下：

a) 操作系统：64 位 Windows 8.1 企业版；
b) 处理器：64 位 Intel Core™ i5-4590 CPU；
c) 内存：8.00GB；
d) GPU：NVIDIA 1080Ti
e) Python IDE：JetBrains PyCharm 2016.3.2(64)；

5.2 文本摘要相关测试

5.2.1 数据集及评价方法

此部分所使用的数据采用了 CCF 在 2017 年举办 NLPCC 比赛时贡献的中文单文档文本摘要数据集 NLPCC2017task#3，这个数据集总共包括 50000 个<原文本，标准摘要>数据对，每对数据中的原文本都是“今日头条”中的新闻报道。实验按照文献[33]中的方式，使用其中的 49500 个数据对对模型进行训练和验证，使用剩余的 500 个数据对对模型的效果进行测试。

模型效果的评价指标采用 Lin 和 Hovy 提出的 ROUGE（Recall-oriented Understudy for Gisting Evalution）评价体系，其中包含多种评价方法，如 ROUGE-*N*，ROUGE-L 等。ROUGE-*N* 的主要思想是统计模型生成摘要和文本标准摘要之间的 *N* 元词的共现情况来对生成摘要的质量进行评估。其中，ROUGE-1 表示生成摘要和标准摘要中有多少个字符是相同的，反映了文本所包含的信息量；当 *N*>=2 时，ROUGE-*N* 反映文本内容的语言流畅性。ROUGE-L 根据 LCS（Longest Common Subsequence,最长公共子序列）算法对生成的摘要进行评估，它更能够从句子级别反映序列的词序。本文使用 ROUGE-1，ROUGE-2 和 ROUGE-L 来评估模型的效果。

ROUGE-*N* 的计算公式如式 5-1 至 5-3 所示。

$$ROUGE-N-R=\frac{\sum_{S\in RefSum}\sum_{g_N\in S}Count_{con}(g_N)}{\sum_{S\in RefSum}\sum_{g_N\in S}Count(g_N)} \tag{5-1}$$

$$ROUGE-N-P=\frac{\sum_{S\in RefSum}\sum_{g_N\in S}Count_{con}(g_N)}{\sum_{S\in GenSum}\sum_{g_N\in S}Count(g_N)} \tag{5-2}$$

$$ROUGE-N-F=\frac{2*ROUGE-N-P*ROUGE-N-P}{ROUGE-N-P+ROUGE-N-P} \tag{5-3}$$

P、*R*、*F* 分别代表准确率、召回率和 *F* 值。式中 *RefSum* 表示标准摘要集合，*GenSum* 表示生成摘要集合，*S* 表示集合中的一条摘要。对于本文使用的数据集而言，原文的标准摘要和生成摘要都只有一个。g_N 代表 *N* 元词，*N* 表示 *N* 元词的长度，$Count(g_N)$表示 *N* 元词的个数，$Count_{con}(g_N)$表示同时出现在标准摘要和生成摘要中的 *N* 元词的个数。

ROUGE-L 的计算方法如式 5-4 至 5-6 所示。

$$ROUGE-L-P=\frac{LCS(r\_s,g\_s)}{n} \tag{5-4}$$

$$ROUGE-L-R=\frac{LCS(r\_s,g\_s)}{m} \tag{5-5}$$

$$ROUGE-L-F=\frac{(1+\beta^2)*ROUGE-L-P*ROUGE-L-R}{ROUGE-L-R+\beta^2*ROUGE-L-P} \tag{5-6}$$

式中 *r_s* 为参考摘要，*g_s* 为生成摘要，*m* 为参考摘要 *r_s* 的长度，*n* 为生成摘要 *g_s* 的长度， *LCS(r_s, g_s)*表示 *r_s* 和 *g_s* 的最长公共子序列的长度，β 是调节因子。

已有研究工作中经常使用 ROUGE 体系中的召回率对模型的效果进行评估，在不加说明的情况下本文也采用此做法。

5.2.2 生成式模型效果评估

此部分主要对本文所提出的两种生成式文本摘要模型进行了相关对比实验设计及结果分析。

本文中使用的带有注意力机制、Coverage 机制和指针生成网络的生成式摘要模型（表示为 A-C-P）的参数设置如下：词向量使用 Word2vec，词向量维度为 256，编码器使用双层的双向 LSTM 实现，解码器使用单层的单向 LSTM 实现，编码器和解码器中隐藏层单元的维度设置为 256，模型词汇表的大小设置为 5 万，模型的优化过程使用 Adagrad 梯度调节算法，*batch_size*，即批处理的大小，设置为 16。模型解码时集束宽度 *beam_size* 的大小设置为 12。

本文提出的基于关键词调整注意力的摘要模型（表示为 Key-Attn）的参数设置与 A-C-P 一致。双编码器摘要模型（表示为 D-Encoder）中子句级别编码器的相关参数设置如下：嵌入层使用 BERT 获得的子句向量，维度为 768，隐藏层使

用单层的双向 LSTM 实现，隐藏层单元维度设置为 256。模型其余部分的参数设置与 A-C-P 一致。

本文设计的对比模型有：本文 3.2 中实现的加入注意力机制的模型（记为 Attention），它主要在解码的每一步使用了注意力机制来获得动态上下文环境向量；在 2017 年 NLPCC 大赛单文档中文摘要生成任务中取得第一名的 NLP_ONE[30]，它不仅在编-解码器之间使用注意力机制，同时在解码器中也使用了注意力机制，以避免生成重复词语；A-C-P；目前在 NLPCC2017task#3 数据集上取得最好效果的模型[33]（记为“主题融合模型”），此模型将关键词信息编码到了解码每一时刻所使用的环境向量中，且使用了 BPE 编码。其中，NLP_ONE 及主题融合模型的结果均来源于文献[33]。表 5-1 展示了各个模型的测试效果。

表 5-1 不同模型测试结果

模型	ROUGE-1	ROUGE-2	ROUGE-L
Attention	0.33801	0.20143	0.28985
NLP_ONE	0.34983	0.21181	0.30686
A-C-P	0.36133	0.22011	0.29796
主题融合模型	0.37667	0.24077	0.32886
Key-Attn	0.37109	0.23307	0.32187
D-Encoder	0.37885	0.24412	0.33116

由 NLP_ONE 与 Attention 的结果可知，在 Decoder 中使用注意力避免生成过多的重复词语可以使得模型效果有所提升。A-C-P 模型由于使用了指针网络，在解码器解码的每一时刻选择性的从原文中复制某些词语以解决 OOV 问题，使得模型的效果得到了很大的提升，且 Coverage 机制可以根据当前解码时刻之前所有解码时刻原文中每个词语的贡献程度对它们进行惩罚，从而避免重复词语的生成，所以此模型效果表现较好。对比主题融合模型和 Key-Attn 可以发现，Key-Attn 比主题融合模型得分略低，这是由于主题融合模型将关键词信息使用一个编码器加入到模型中，且在此编码器上使用注意力机制，即其关键词的注意力是动态的，而本文虽然使用有监督方法获得原文中每个词语属于关键词的概率，但这种概率却是静态的。此外，D-Encoder 得分比主题融合模型略高，这说明子句级别的编码器在解码的每一步获得原文中每个子句对当前预测词语的贡献程度然后对原文信息进行编码可以在一定程度上获得更加准确的上下文向量。最后，对比主题融合模型、Key-Attn、D-Encoder 与其余模型可以看出，在模型中引入额外信息，如关键词信息和子句信息，可以使得模型的效果在一些经典模型基础上有进一步的提升。

为了更好的呈现各个模型的效果，文本对 Attention、主题融合模型、Key-Attn 及 D-Encoder 生成的摘要进行了一个展示，具体的见表 5-2 所示。

表 5-2 生成摘要对比示例

标准摘要	昨日下午，山西平遥县6名儿童结伴滑冰玩耍时，不慎溺水身亡，其中年龄最大的11岁，最小的为5岁。
Attention	山西‘UNK’‘UNK’发生发生一起意外溺水事件，6名儿童不幸溺水身亡。
主题融合模型	组图：平遥县6名儿童结伴滑冰玩耍不慎落水，其中年龄最大的11岁，最小的为5岁，最小的为5岁
Key-Attn	平遥县6名儿童结伴滑冰时溺水身亡，其中年龄最大的为11岁，最小的5岁
D-Encoder	昨日下午，平遥县6名儿童结伴滑冰玩耍时溺水身亡，年龄最大的为11岁，最小的为5岁。
标准摘要	达州一煤矿发生瓦斯爆炸事故4人被困井下，1人受伤，相关部门正在全力救助被困人员。
Attention	今日四川省煤矿改造发生‘UNK’事故，造成4人被困，川煤集团
主题融合模型	组图：达川区发生瓦斯爆炸事故，4人被困井下，1人受伤，伤者已送达州医院救治。
Key-Attn	达州一煤矿发生瓦斯爆炸事故，造成4人被困井下，救援受伤，无生命危险。
D-Encoder	达州开发区工程发生瓦斯爆炸事故，致4人被困井下，救援仍无生命危险

由表中举例可以看出，生成式模型通过大量数据学习人类撰写标题的习惯，它不是将文中部分内容进行抽取和拼凑来简单的生成摘要，而是在理解全文语义及主题的基础之上，生成更加连贯、流畅、符合人类阅读习惯的文本内容。对比 Attention 与其余三个模型可以看出，此模型对于全文的重点信息并没有把握完整，且由于没有使用指针网络，模型所生成的摘要中含有未登录词（即“UNK”）。对比主题融合模型和 Key-Attn、D-Encoder 可知，虽然主题融合模型在解码时使用了解码注意力以指导解码过程，但其生成的结果中仍存在重复内容，而本文模型中使用了 Coverage 机制，较好的解决了这一问题。最后，对比本文提出的两个模型的生成内容与标准摘要可以看出，模型生成的摘要注重对于事件信息的表达，但内容的准确性和完整性仍有欠缺。

5.2.3 双阶段模型效果评估

双阶段式摘要模型中第二阶段所使用的是带注意力机制、Coverage 机制以及使用指针网络的生成式文本摘要模型，其具体参数设置与 5.2.2 节中一致。

本文首先对第一阶段抽取文本关键内容的各个做法中所涉及的参数设置做了相关的测试和说明。主要包括获得子句语义得分时制作训练集选取 top*Kss* 个子句标记为“1”时 *Kss* 值的设置以及对每个子句的语义得分进行调整时 MMR 算法中 λ 值的设置。

对于 *Kss* 取值的确定，本文设计的实验如下：首先设置 *Kss* 取不同的值，然后根据 3.3 节步骤 a)中计算得到的每个子句与标准摘要的相似度，取相似度最高的 *Kss* 个子句作为第二阶段生成式摘要模型的输入进行训练并测试，然后根据测试结果确定 *Kss* 的取值。当 *Kss* 取不同值时，测试结果如表 5-3 所示。

表 5-3 *Kss* 取值的测试

Kss 取值	ROUGE-1	ROUGE-2	ROUGE-L
5	0.33417	0.19197	0.27894
7	0.34868	0.21009	0.29147
9	0.35972	0.21913	0.30296
11	0.36064	0.21981	0.30284
13	0.36015	0.21947	0.30289

由表中数据可知，随着 *Kss* 值的增大，模型效果先提升然后不再有明显变化，这说明，当 *Kss* 的值设置过小时，则认为只有与标准摘要高度相似的少量语句才是文本的关键内容，但由于文本中可能存在多个语义相同的子句，当它们都与标准摘要所表达的内容高度相似时，则选择的内容对于文本信息的覆盖度就较弱，导致模型效果较差。随着 *Kss* 值逐渐增大，选择的内容中逐渐包含更多的语义信息，但随着 *Kss* 继续增大，选择的内容中包含的关键信息不再增多，模型效果不再有明显变化。所以最终本文设置 *Kss* 的取值为 11，即将文本中与标准摘要相似性最高的 11 个子句标记为“1”，其余子句标记为“0”，然后对子句语义得分预测模型进行训练。

MMR 算法中参数 λ 的确定方法是，分别设置 λ 为 0.9、0.7、0.5、0.3 和 0.1，对原文中每个子句的语义得分进行调整获得子句的最终语义得分，然后根据此得分抽取出得分最高的几个子句作为生成式摘要模型的输入进行测试。由于 MMR 算法去除了部分冗余子句，所以此处选取出的子句数应该不多于 11 条，否则可能造成虽然 λ 取值不同，但最终抽取出的子句却是相同的。最终通过人工对数据进行抽样分析，决定抽取出的子句数 *K* 的确定方式如式 5-7 所示。

$$K=\begin{cases}5，\text{文本子句数不多于 } 15 \text{ 条}\\ 7，\text{文本子句数不多于 } 25 \text{ 条}\\ 9，\text{文本子句数大于 } 25 \text{ 条}\end{cases} \tag{5-7}$$

λ 不同取值时，模型测试结果如表 5-4 所示。

表 5-4 λ 取值的测试

λ	ROUGE-1	ROUGE-2	ROUGE-L
0.9	0.35367	0.21145	0.30047
0.7	0.36514	0.22643	0.30781
0.5	0.35865	0.21595	0.29828
0.3	0.32771	0.19897	0.27696
0.1	0.27079	0.15496	0.23512

由测试结果可知，当 λ=0.7 时，模型的效果最好。此时抽取出的子句所包含的信息既与标准摘要所表达的主题高度相似，且文本内容所表达的语义也更加丰富，在降低文本长度的同时保证了模型效果。

其次，本文对确定每个子句最终得分时，式 3-28 中 α 值的选取进行了测试。模型最终为每条文本抽取出的子句数 K 的确定方式与式 5-7 表达的一致。测试结果如表 5-5 所示。

表 5-5 α 取值的测试

α	ROUGE-1	ROUGE-2	ROUGE-L
1.0	0.36514	0.22643	0.30781
0.8	0.37176	0.23156	0.31365
0.6	0.36347	0.22579	0.30843
0.4	0.35012	0.21215	0.30016
0.2	0.33141	0.19357	0.27435

由测试结果可以看出，当 α 值过大或者过小时，模型效果都不好，所以在抽取子句时，如果考虑的因素过少，则会导致抽取的内容不适当。此外，当 α 为 0.8 时，模型效果最好，这说明相比于关键词得分，语义得分更能判断子句的重要性，但同时由于指针网络的使用，子句的关键词含量问题也不容忽视。

最后，本文将本文提出的模型与其他几种双阶段模型进行了对比，所有双阶段模型在第二阶段所使用的生成式模型相同。第一阶段，即文本关键内容的抽取，本文设计了以下几种方式：借鉴文献[4]的研究结果，直接使用原文中前 K-1 条子句及最后一条子句作为抽取内容（表示为 First_Last）；DK_TextRank [71]，此算法先使用 Doc2Vec 进行文本向量化并使用 K-Means 算法对文本子句进行聚类，然后在各个类簇中使用 TextRank 算法对子句进行打分，最终从每个簇中选择子句作为抽取出的内容；基于 TF-IDF 算法提取文本关键词，然后根据子句中关键

词的含量获得子句得分并选取出得分最高的 K 条子句。（表示为 TF-IDF）；本文做法，表示为 ECSS(Extarct Content based on Supervised Scoring Algorithm)。此外，对比试验中还加入了不对原文本进行任何过滤，直接使用第二阶段生成式模型的做法，表示为 ORIG。测试结果如表 5-6 所示。

表 5-6 双阶段模型效果对比

模型	ROUGE-1	ROUGE-2	ROUGE-L
First_Last	0.34286	0.20564	0.29115
DK_TextRank	0.35641	0.21637	0.29943
TF-IDF	0.36368	0.22615	0.30412
ORIG	0.36133	0.22011	0.29796
ECSS	0.37176	0.23156	0.31365

由测试结果可以看出，虽然有研究表明文本的关键内容位于段首的可能性高达 85%，位于段尾的可能性达 7%，但对于本文所使用的数据集而言，直接使用文首 K-1 条子句及最后一条子句的内容进行摘要生成的效果较差。DK_TextRank 算法没有使模型获得较高的提升，这是因为此算法对文本进行主题聚类后，在每个簇中都进行选取子句，但实际上有些簇所代表的主题与摘要信息联系并不紧密，造成选取内容不准确。TF-IDF 算法获得了与使用原文相当的效果，这说明文本关键词信息对原文主题的把握至关重要。此外，本文方法比直接使用原文进行生成式摘要的效果在 ROUGE-1、ROUGE-2、ROUGE-L 上分别提升了 1.0%、1.1% 和 1.6%，而其余方法均没有得到很好的效果，这是由于本文所提出的方法使用了有监督算法获得子句的语义得分以及关键词得分，即本文所抽取出的内容对于标准摘要而言更具有针对性。

本文对各种抽取算法所抽取出的内容也进行了分析，分别统计了抽取出文本的平均词语数（记为 num_words）、文本包含标准摘要中词语的平均数量（记为 num_keys）以及文本中属于标准摘要中词语的数量占文本总词语数的平均比例（记为 ratio_src）。各抽取方法统计结果见表 5-7 。

表 5-7 关于各抽取方法抽取出内容的统计

抽取方法	num_words	num_keys	ratio_src
ORIG	574.53	17.08	0.0653
First_Last	182.64	12.83	0.1039
DK_TextRank	201.72	11.29	0.0927
TF-IDF	317.61	15.46	0.0855
ECSS	257.81	14.76	0.0958

由统计结果可以看出，虽然与 First_Last、DK_TextRank 方法相比，本文抽取内容的文本长度更长，但其内容中含有的标准摘要中的词语更多。TF-IDF 方法由于从关键词的角度对子句打分，所以抽取内容中含有较多的文本关键词，则含有的标准摘要中的词语也较多，但其抽取出的文本内容较长。与原文相比，本文方法在对含有标准摘要中词语量减少较少的情况下，降低了文本长度，减少了模型的时空资源消耗，且保证了第二阶段生成式模型的效果。

5.3 文本分类相关测试

5.3.1 数据集及评价方法

本文文本分类部分所使用的数据集是根据关键字从网络平台中爬取得到的，按照 4.1 节中的方式对数据进行处理后，每种类别数据的数据量如表 5-8 所示。本文将数据按照 7:2:1 的比例进行划分，分别用作训练、验证和测试过程。

表 5-8 各类警情数据的数量

类别	数据量	类别	数据量
偷盗类	889	毁坏财物类	71
两抢类	220	扰乱公共秩序和危害公共安全类	183
人身伤害类	696	交通事故类	821
黄赌毒类	135	维权类	1007
诈骗类	922	安全事故类	301
传销类	138		

为了对模型的分类效果进行评估，本文采用文本分类任务中的 P-R-F 评价体系对模型的测试结果进行评价。其中 P 代表精确率（Precision），R 代表召回率（Recall），F 代表 F1 得分（F1-score），其各自的定义如下：

$$P = \frac{TP}{TP+FP} \tag{5-8}$$

$$R = \frac{TP}{TP+FN} \tag{5-9}$$

$$F = \frac{2*Recall*Precision}{Recall+Precision} \tag{5-10}$$

假设类别标签为 A 和 B，则上式中的 *TP* 表示样本的真实类别为 A 而模型预测得到的样本类别也为 A 的样本数；*FP* 表示样本的真实类别为 B 而模型预测得到的样本类别为 A 的样本数；*FN* 表示样本的真实类别为 A 而模型预测得到的样本类别为 B 的样本数。

精确率 P 代表模型判断为 A 类的数据中样本确实为 A 类的概率，它可以评估模型预测结果的可信度；R 代表 A 类数据中有多少比例的数据被模型预测为了 A 类，它可以评估模型的查全率；F 值从权衡 P 和 R 的角度说明模型预测能力的平稳性。

P-R-F 评价体系有“micro”、“macro”和“weight”三种模式，分别从不同角度对模型的测试结果进行了评估。在不加说明的情况下，本文中均使用模型在所有类别上测试结果的平均值，即使用“macro”模式来展现模型的测试效果。

5.3.2 效果评估

模型的参数对模型的学习效果有很大影响，所以本文首先对 SPCNN 模型中分类任务所使用的预训练模型 TextCNN 中的参数进行了相关测试，分别从字向量的维度、卷积窗口的大小以及卷积核的数目对 TextCNN 最终分类效果的影响进行了实验。其中，字向量的维度和卷积核的数目都设置了 32、64、128、256、512 和 1024 这 6 组对比实验，卷积窗口的大小设置了[4]、[5]、[6]、[4,5] ,[5,6]，[4,6]和[4,5,6]这 7 组对比试验，[4,5]表示同时使用窗口大小为 4 和 5 的卷积核进行卷积操作，其余类似。实验结果如图 5-1 至 5-3 所示。

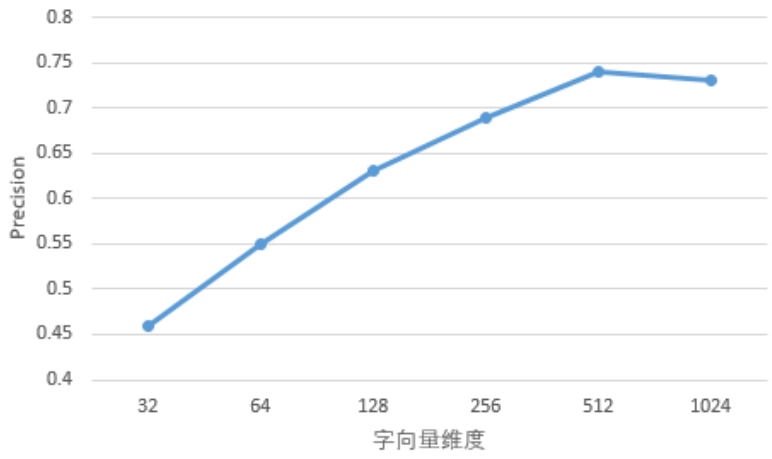

图 5-1 字向量维度对分类效果的影响

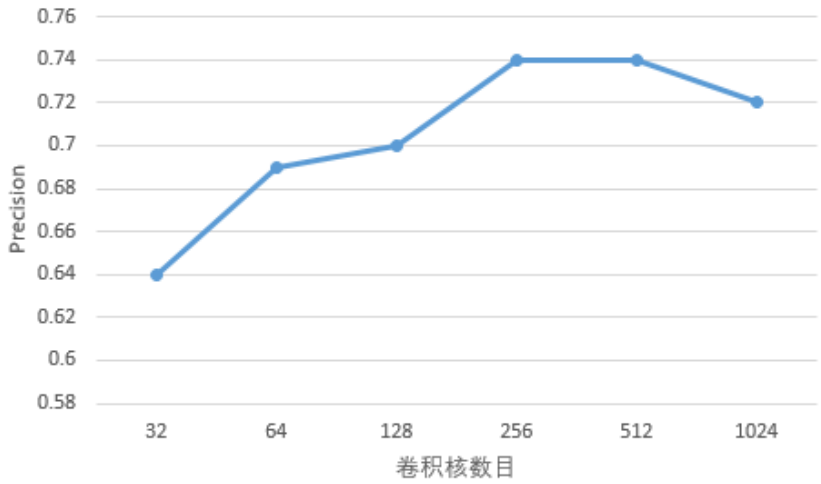

图 5-2 卷积核数目对分类效果的影响

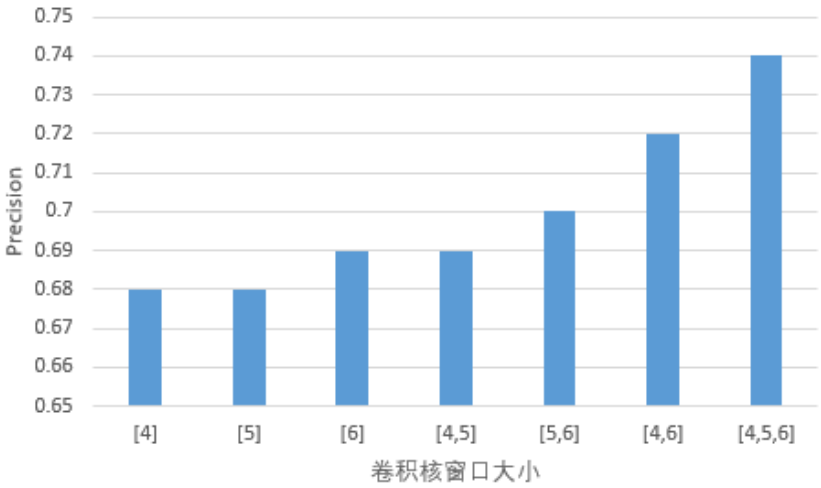

图 5-3 卷积核窗口大小对分类效果的影响

由图 5-1 可以看出，随着字向量维度的增加模型的准确率先提升然后降低，这说明字向量具有一定的语义表征能力，但并不是使用维度越高的字向量模型的效果就越好，因为维度太高会使得模型理解语义困难，从而导致模型提取特征困难，模型效果变差。从卷积核数目的测试结果来看，当卷积核数目较少时，模型效果不佳，这是因为模型提取到的数据特征较少，不足以表达文本的语义，随着卷积核数目的增多，模型提取到的特征逐渐丰富，但最后模型效果却不再提升，因为此时模型已经过拟合。最后，由图 5-3 看出，使用多种窗口大小的卷积核进行卷积操作可以提取到更加丰富的数据特征，模型的分类效果更好。

三个实验的具体测试结果见表 5-9。本着尽量减少模型参数、降低模型资源消耗的原则，本文最终确定 SPCNN 模型每层网络的特征提取子网络中各超参数的设置如下：输入矩阵的列数，即字向量的维度，为 512；输入数据矩阵的行数，

根据 4.1 节中表 4-1 中步骤 3 的方式确定；卷积核的尺寸为[4,5,6]；卷积核的数目为 256；池化层采用 Max Pooling。

表 5-9 不同参数下模型效果

字向量维度	P/R/F	卷积核数目	P/R/F	卷积窗口大小	P/R/F
32	0.46/0.62/0.52	32	0.64/0.72/0.67	[4]	0.68/0.73/0.69
64	0.55/0.66/0.59	64	0.69/0.76/0.72	[5]	0.68/0.75/0.71
128	0.63/0.7/0.65	128	0.7/0.75/0.72	[6]	0.69/0.75/0.71
256	0.69/0.75/0.71	256	0.74/0.79/0.76	[4,5]	0.69/0.76/0.72
512	0.74/0.79/0.76	512	0.74/0.77/0.75	[5,6]	0.70/0.76/0.73
1024	0.73/0.78/0.75	1024	0.72/0.76/0.74	[4,6]	0.72/0.78/0.75
				[4,5,6]	0.74/0.79/0.76

其次，本文测试了预训练操作对网络最终分类效果的影响。由于 SPSN 网络的预训练需要用到每层网络中特征提取子网络 FESN 得到的特征图，即网络提取出的数据特征，则当对 SPSN 进行预训练前需要保证图 4-1 中 c1 所对应的网络也已经被预训练，所以本文设置了三组对比试验：FESN 网络和 SPSN 网络均不进行预训练（表示为 not_pre）、只对 FESN 网络进行预训练(表示为 pre_FESN)、既对 FESN 网络进行预训练又对 SPSN 网络进行预训练(表示为 pre_FESN_SPSN)。

由于本模型中第二层网络的输入内容更具有表征性，所以本文使用第二层网络的分类结果代表文本所提出模型的效果（记做 SPCNN_2）。此外，由于特征融合的优点，所以本文对特征融合后的分类效果也进行了测试(记做 SPCNN_12)。测试结果如表 5-10 所示。

表 5-10 预训练对 SPCNN 模型的影响

	SPCNN_2			SPCNN_12		
	P	R	F	P	R	F
not_pre	0.78	0.80	0.78	0.79	0.81	0.79
pre_ FESN	0.79	0.80	0.79	0.80	0.81	0.79
pre_FESN_SPN	0.82	0.81	0.80	0.84	0.83	0.82

由表中测试结果数据可以看出，当对网络中的特征提取子网络进行预训练时，相比于不进行预训练，无论是 SPCNN_2 还是 SPCNN_12，测试结果只在准确率或者 F 值上提高了 1 个百分点，所以特征提取子网络是否进行预训练对模型最终的效果影响并不大。对 SPSN 进行了预训练后，与不对其进行预训练的效果相

比， SPCNN_2 的测试结果在 P、R、F 上分别比高了 3%、1%和 1%， SPCNN_12 的测试结果在 P、R、F 上分别提升了 4%、2%和 3%，这说明预训练操作对 SPSN 网络预测到原序列中更具表征性的子序列有很大作用。最后，通过对比 SPCNN_2 与 SPCNN_12 的结果可以看出，特征融合后的效果比单独使用一种特征的效果提高了 1 至 2 个百分点，这说明特征的融合能在一定程度上使模型的分类效果得到优化。

在对 SPSN 网络进行预训练时，需要有一个参考子序列供 SPSN 网络学习以输出更加正确的 t_y 和 t_l，参考子序列的中心 t_y^*可以根据本层网络提取到的特征中的最大响应值获得，而对于 t_l^*，本文通过将其设置为一个固定值来实现。不同取值的 t_l^*会对 SPSN 网络训练的结果造成不同的影响，从而对模型的分类效果造成影响，所以本文对不同 t_l^*的取值对模型分类效果的影响进行了测试，测试结果如表 5-11 所示。

表 5-11 SPSN 预训练中t_l^*取值对 SPCNN 模型效果的影响

t_l^*	SPCNN_2			SPCNN_12		
	P	R	F	P	R	F
0.20	0.60	0.50	0.47	0.76	0.80	0.77
0.25	0.76	0.77	0.75	0.77	0.80	0.78
0.30	0.77	0.79	0.78	0.81	0.82	0.80
0.35	0.82	0.81	0.80	0.84	0.83	0.82
0.40	0.79	0.81	0.79	0.81	0.81	0.80

由表中的结果看出，随着 t_l^*的增加，模型的效果有了很大的提升，但是当 t_l^*的值大于 0.35 之后，模型的效果便开始下降了。这说明当 t_l^*=0.2 时，模型第二层网络输入对应的文本所包含的信息量过少，使得模型提取不到充分的数据特征。随着 t_l^*的增大，SPSN 网络逐渐定位到文本中包含信息更加丰富和集中的内容区域，使得模型提取到的特征更加准确，模型分类效果更优。

当 t_l^*=0.30 和 t_l^*=0.35 时 SPSN 网络截取的数据所对应的文本内容见表 5-12 所示。

表 5-12 t_l^*=0.30 和 t_l^*=0.35 时 SPSN 网络截取内容示例

原文	四月四号 ，早上八点二十左右 ，在成都地铁四号线 b 出口，成温立交桥红绿灯的路口上，两个小伙子，穿黑色 T 恤，骑 摩托车，在等红绿灯的时候，后面的那个小伙子下车了，我以为是过马路，协警会要求不能载人，所以他下来了，结果他看了两下，看见有一个女士背着双肩包，把手机放着书包的左侧放水杯的那个兜了，插着耳机，正在听歌，结果他直接扯出来，把耳机扯掉，上摩托车就跑了。现在小偷竟然如此猖獗！……
t_l^*=0.3	插着耳机，正在听，结果他直接扯出来，把耳机扯掉，上摩托车就跑了。现在小偷竟然如此猖獗！……
t_l^*=0.35	手机放着书包的左侧放水的那个兜了，插着耳机，正在听，结果他直接扯出来，把耳机扯掉，上摩托车就跑了。现在小偷竟然如此猖獗！……
原文	1 月 13 日晚上 10 点，众多网友爆料称在天府大道南三段发生一起交通事故，一辆劳斯莱斯轿车与三轮车相撞，现场交通拥堵，伤亡情况不详。随即，成都商报客户端记者从成都市交警七分局了解到，晚上 7 点 10 分左右，在天府大道南三段发生一起交通事故，一辆白色劳斯莱斯轿车从仁寿往成都方向行驶，与一辆横穿天府大道的三轮车相撞，事故已导致 2 人死亡。事故发生后，交警部门立即达到现场调查处理，并疏导交通，现交通已恢复正常，劳斯莱斯轿车驾驶员被带回进行调查，通过初步检测，暂没发现驾驶员有酒驾，最终结果仍需等待抽血检测。
t_l^*=0.30	事故发生后，交警部门立即达到现场调查处理，并疏导交通，现交通已恢复正常，劳轿车驾驶员被带回进行调查，通过初检测，暂没发现驾驶员有酒驾，最终结果仍需等待抽血检测。
t_l^*=0.35	三轮车相撞，事故已导致 2 人死。事故发生后，交警部门立即达到现场调查处理，并疏导交通，现交通已恢复正常，劳轿车驾驶员被带回进行调查，通过初检测，暂没发现驾驶员有酒驾，最终结果仍需等待抽血检测。

由表 5-12 中的数据可知，相比于 t_l^*=0.30，t_l^*=0.35 时，网络截取到的文本内容所包含的每个类别的关键信息更加丰富，如对于第一条偷盗类数据，t_l^*=0.35 时，网络多截取到了“手机”一词，第二条人身伤害类数据，网络多截取到 “相撞”和“死”两个关键词语。

当 t_l^*=0.35 和 t_l^*=0.40 时 SPSN 网络截取的数据所对应的文本内容见表 5-13 所示。

表 5-13 t_l^*=0.35 和 t_l^*=0.40 时 SPSN 网络截取内容示例

原文	今天上午，在成都市量力钢材城外，记者见到了刚卸完货的拖挂车司机裴师傅和尹师傅，……，显得特别打眼，除了挂车的主油箱油被洗劫一空外，就连自备的油罐里的油也被盗走了。发现油被盗后，裴师傅赶紧去加油站买了点油回来，才将车开进了钢材城卸货。裴师傅说，他们从河北沧州市两人交替开车，颠簸了近 40 个小时，才来到了成都。由于路途较远，加之各地油价不同，所以他们每次临走前都会在油罐中备足油。
t_l^*=0.35	发现油被盗后，师赶紧去加油站买了点油回来，才将车开进了钢材城货。师说，他们从河北市两人交开车，了近 40 个小时，才来到了成都。由于路途较远，加之各地油价不同，所以他们每次临走前都会在油中备足油。
t_l^*=0.40	就连自备的油罐里的油也被盗走了。发现油被盗后，师赶紧去加油站买了点油回来，才将车开进了钢材城货。师说，他们从河北市两人交开车，了近 40 个小时，才来到了成都。由于路途较远，加之各地油价不同，所以他们每次临走前都会在油中备足油。
原文	位置于成都西区医院（二环路西三段北）这家咖啡馆有酒托就是下面那个女的，这女的专门上各种相亲网上结识网友会用各种行骗技巧进行行骗，这家商铺就是一家合伙搭子，里面的服务态度很差，尤其是会对外地人员下手，外地人很吃亏，所以希望大家看到此内容请转发，不要让这样的害群之马坏了一锅粥，不要再让这种恶劣事情污染的城市，谢谢大家
t_l^*=0.35	会用各行骗技进行行骗，这家商铺就是一家合伙搭子，里面的服务态很差，尤其是会对外地人员下手，外地人很吃，所以希望大家看到此内容请转发，不要让这样的害群之马坏了一，不要再让这恶事情污的城市，谢谢大家”
t_l^*=0.40	酒托就是下面那个女的，这女的专门上各种相亲网上结识网友会用各行骗技进行行骗，这家商铺就是一家合伙搭子，里面的服务态很差，尤其是会对外地人员下手，外地人很吃，所以希望大家看到此内容请转发，不要让这样的害群之马坏了一，不要再让这恶事情污的城市，谢谢大家”

由表 5-13 中的数据可以看出，相比与 t_l^*=0.35，t_l^*=0.40 时，SPSN 网络所截取到的内容中会含有少量的冗余成份或者无用信息。如第一条数据中的“就连自备的油罐里的油也被盗走了”，由于后文内容中已经含有“发现油被盗了”这一关键信息，所以此时再添加这一内容反而会增加文本长度，而不能使模型效果得到提升。而第二条数据中 “酒托就是下面那女的…”这一内容对于“诈骗类”数据并没有明显的辨别作用，所以添加此内容反而会增加模型对于数据语义信息的理解难度，使得模型效果下降。

之后，本文将提出的模型与几个机器学习算法以及深度学习算法进行了对比。

SVM 表示基于 TF-IDF 特征的支持向量机模型；SVM_Key 表示基于类别关键词特征的支持向量机模型，具体做法是：首先对每类数据统计词频，然后将只属于此类别高频词而不属于其他类别高频词的词语选为该类别的关键词，统计出的各类别的关键词集合如表 5-14 所示，然后将这些关键词集合的并集作为模型的词汇表，并使用词袋模型的思想将数据集中的每条数据向量化，最后根据向量化后的数据及其对应的标签使用 SVM 模型进行训练；NB 表示基于词袋模型的多项式贝叶斯模型；TextCNN，SPCNN 模型每层网络分类时所使用的结构，即 FESN 网络结合 CSN 网络；RCNN[42]，它是将 RNN 和 CNN 进行结合设计出的模型。各个算法在测试集上的效果对比如图 5-4 所示。

表 5-14 类别关键词

类别	关键词
偷盗类	偷、盗、监控、窃、团伙、贼、入室、包包、手机
两抢类	抢、劫、夺、飞车、手机
黄赌毒类	毒、麻、农家乐、贩卖、吸食、甲基苯丙酸、尿液、赌、输、赢、麻将、淫、嫖、黄、开房、色情、小姐、酒店、
交通事故类	车祸、肇事、逃逸、当场、受损、变形、救援、事故、惨烈、避让、驾驶员、撞、司机、路口、追尾、醉驾
人身伤害类	杀、伤害、受害、故意、打、强奸、猥亵、坠楼、被害、嫌疑、威胁、故意
传销类	推销、组织、洗脑、介绍、传销、直销、返利、打击、发展、生意
安全事故类	人员伤亡、消防、燃、火、扑灭、扑救、突发、浓烟、灾
诈骗类	装修、贷款、健身、资金、电话、消费者、老年人、购、平台、买、培训、朋友、充值、会员、诈、微信、骗、跑路、招聘、公司
毁坏公共财物类	经济、烧毁、共享、损失、打砸、焚烧、破坏、损坏、单车、
维权类	代表、解决、土地、业主、维权、商家、投诉、社区、拆迁、合同、房、教师、家长、居民、百姓、村民、开发商
扰乱公共秩序和危害公共安全类	谣言、飙车、扰、无人机、公共、飞、黄牛、行政拘留、航、秩序、备降、机场

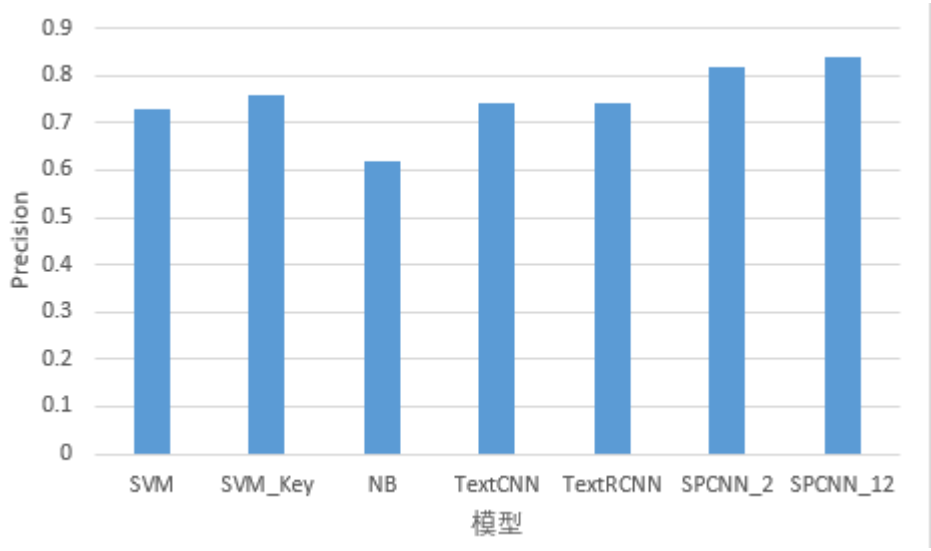

图 5-4 各分类模型测试准确率

各个模型具体的测试结果如表 5-15 所示。

表 5-15 不同分类模型测试结果

模型	P	R	F
SVM	0.73	0.71	0.69
SVM_Key	0.76	0.74	0.73
NB	0.62	0.70	0.64
TextCNN	0.74	0.79	0.76
RCNN	0.74	0.78	0.76
SPCNN_2	0.82	0.81	0.80
SPCNN_12	0.84	0.83	0.82

由图 5-4 及表 5-15 中的数据可以看出，本文提出并实现的文本分类模型 SPCNN 第二层网络的测试效果与其他模型相比，在测试集上达到了较高的准确率、召回率以及 F 值，在准确率上比 SVM_Key 高了 6 个百分点，在召回率上比 TextCNN 高了 2 个百分点，在 F 值上比 TextCNN 高了 4 个百分点。而特征融合后，模型效果又进一步得到了提升，在 P、R、F 上均提升了 2%。SVM_Key 相比于 SVM 在准确率、召回率以及 F1 得分上分别提升了 3 个百分点、3 个百分点和 4 个百分点，这说明人工提取出的特征相对于通用特征而言更能表达数据的特点，从而使得模型更能准确的识别出不同类别的数据。但是，人工提取特征一方面需要耗费大量时间和精力，另一边方面，提取的特征需要根据数据集的不同而进行及时调整，过程繁琐。此外，RCNN 模型的效果并不优于 TextCNN，说明在本文数据集上，将序列结构加入卷积神经网络中并不能提高模型对文本的语义理

解能力。最后，本文 SPCNN_12 的效果比单独使用一层 CNN 网络的 TextCNN 模型的效果提高了较多，这证明了本文模型的优越性。

5.4 本章小结

本章首先说明了本文实验测试部分所基于的实验环境，然后分别对本文所提出的文本摘要模型和文本分类模型从不同的角度进行了测试并与领域内的其他模型进行了效果对比。

第六章 总结与展望

6.1 论文总结

随着国家经济的发展和当今时代互联网技术的普及，人们逐渐的使用电子设备通过网络通道进行日常的交流、工作和获取信息等。各种网络平台及社交媒体的快速发展使得广大网民成为了网络舆情信息传播的主要介质，网络的这种开放性及虚拟性一方面使得任何人在任何地方都能平等、自由的发表自己的观点，但另一方面使得网络中存在海量杂乱无章的舆情信息。这些杂乱、冗长甚至带有个人感情色彩的舆情报道信息一方面会给广大网民带来困扰，另一方面也会给舆情分析人员带去困难。为此，本文研究了文本摘要技术和文本分类技术，以对网络中舆情数据进行处理和分析。文本主要研究及工作如下：

a) 结合当前网络舆情数据的特点，针对其信息过于零散、信息量过大问题，对舆情分析工作所涉及到的自然语言处理中的相关技术及相关实现方式进行了调研和分析。

b) 针对舆情信息中信息量大、新闻报道的文本过长及读者时间精力有限的问题，对文本摘要技术进行了研究，并针对目前模型中所存在的一些问题，提出了三种改进方法。第一种是根据文本的关键词信息对 SeqSeq 模型中的注意力机制进行改进，使得模型对于关键词语更加敏感；第二种是针对目前大多数模型都是从原文的词级别或者字符级别对原文进行编码的问题提出了一种双编码器模型，模型同时从词级别和子句级别对原文进行编码以使得解码器使用的语义向量更加准确；最后一种是针对原文本过长时模型效果不佳的问题，提出了一种双阶段式文本摘要模型，通过将原文中的关键内容抽取出作为生成式摘要模型的输入序列提高模型的效果。

c) 针对舆情信息过于零散，难以从中获得高层次有价值信息的问题，本文以成都地区公安机关对警情信息处理的相关需求为例进行了相关研究。首先，根据定义的警情类别为每个类别设置线索词，根据此线索词使用爬虫技术从各个网络平台中获得了警情数据，随后对数据进行了相关过滤并人工对数据类别进行标注，形成了一个标准的分类数据集。然后对数据进行分析，并根据警情数据的特点基于 CNN 网络设计并实现了一种新颖的文本分类模型，在警情数据分类上达到了较好的效果。

d) 对于以上提出的模型，本文首先对其中涉及到参数设计进行了相关测试，然后设计对比实验，在公开数据集上或者自己准备的专业领域数据集上，使用已有的评价方法对其结果进行对比和评估。

6.2 展望

在本文的研究、实现及测试过程中，本文发现本文的研究工作中还存在如下问题需要解决：

a) 本文在文本摘要模型中对词进行嵌入时使用的是较为基础的 Word2vec 词向量，此词向量对于多义词只能生成一个向量，这样会增加模型对文本语义理解的困难。使用其他更具有表征能力的词向量提高模型效果是本文需要继续研究的。

b) 在本文文本摘要模型实现中，分词处理使用的是 jieba 工具，系统词汇表的大小设置为 5 万，但由于目前网络用语的流行，jieba 分词效果并不好，且词汇表较小也导致较多的未登录词及罕见词，所以本文拟在下一步使用 BPE 弱化此问题。

c) 本文提出的文本分类模型 SPCNN 取得最好效果是在使用 ti^{*}=0.35 对 SPSN 网络进行预训练时，此时相当于提取出的子序列占原序列的 70%，如何进一步减小此比例或者能否使模型预测多个更加简短的子序列然后进行分类是本文下一步需要进行研究的。

致谢

从 2017 年夏季到现在，我已经在电子科技大学度过了将近三年的研究生时光。在这三年里，我经历过初至此地的兴奋与紧张，经历过身心成长的喜悦与孤单，经历过学习研究中的迷茫与挫败。在这即将与这里的一切告别的时候，心中有许多感慨，借此时机向这些在我人生成长短途中给予过我陪伴和帮助的人表示感谢。

首先要感谢的是我的导师***。在这三年的学习与研究中，老师总是在第一时间给予我学习方向上的指导，做我学业上的指明灯，并且会经常纠正我在学习中的态度问题，告诫我既要将理论知识掌握好也要将实践能力提升上来。此外，老师还会在生活上对我给予关心，让我感受到了温暖。

其次要感谢的就是我的家人和朋友们。虽然身在远方，但是家人们对我的关心却如在身边，哥哥、嫂子对我生活上的关心、小侄女对我心理上给予的支持以及爸妈对我的牵挂让我在每一次失落的时候倍感欣慰。感谢王小花、谢小布每一次的心理开导，让我的生活多一道曙光。感谢龙龙、克哥的每一次谈心，让我感到放松。感谢实验室的小伙伴们，与我共同进步和奋斗。

最后要感谢参与本文评审工作及提出相关意见的老师，也对本文研究工作中参考过其论文、博客、实践工作的前辈们表示感谢。

参考文献

[1] 唐喜亮. 我国突发公共事件的网络舆情研究[D]. 成都, 电子科技大学, 2008, 1-60.

[2] 杨欢. 基于文本分类的微博情感倾向研究[D]. 重庆, 重庆师范大学, 2016, 1-59.

[3] H. P. Luhn. The Automatic Creation of Literature Abstracts[J]. IBM Journal of Research and Development, 1958, 2(2): 159-165.

[4] P. B. Baxendale. Machine-Made Index for Technical Literature-An Experiment[J]. IBM Journal of Research and Development, 1958, 2(4):354-361.

[5] G. Salton, C. T. Yu. On the Construction of Effective Vocabularies for Information Retrieval[C]. Proceedings of the 1973 International ACM SIGIR Conference on Research and Development in Information Retrieval. Gaithersburg, 1973, 48-60.

[6] J. Kupiec, J. O. Pedersen, F. Chen. A trainable document summarizer[C]. Proceedings of the 18th International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, 1995, 68-73.

[7] G. E. Hinton, R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks[J]. Science, 2006, 313(5786):504-507.

[8] Y. Liu, S. H. Zhong. Query-oriented multi-document summarization via unsupervised deep learning[C]. Proceedings of the 26th AAAI Conference on Artificial Intelligence. Toronto, 2012, 1699-1705.

[9] A. M. Rush, S. Chopra, J. Weston. A Neural Attention Model for Abstractive Sentence Summarization[C]. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, 2015, 379-389.

[10] K. Cho, B. Merrienboer, C. Gülçehre, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[C]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, 2014, 1724-1734.

[11] D. Bahdanau, K. Cho, Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate[OL]. https://arxiv.org/pdf/1409.0473v2.pdf.

[12] Y. Bengio, R. Ducharme, P. Vincent, et al. A Neural probabilistic language Model[J]. Journal of machine learning research, 2003, 3(Feb):1137-1155.

[13] C. Y. Lin. ROUGE: A Package for Automatic Evaluation of summaries[C]. Proceedings of the Workshop on Text Summarization Branches Out, Post-Conference Workshop of Association for Computational Linguistics (ACL), Barcelona, 2004, 74-81.

[14] S. Chopra, M. Auli, A. M. Rush. Abstractive sentence summarization with attentive recurrent

neural networks[C]. Proceedings of the 2016 conference on North American Chapter of the Association for Computational Linguistics, San Diego, 2016, 93-98.

[15] R. Nallapati, B. Zhou, C. N. Santos et al. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond[C]. Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning(CoNLL), Berlin, 2016, 280-290.

[16] K. M. Hermann, T. Kociský, E. Grefenstette, et al. Teaching machines to read and comprehend[C]. Proceedings of the 2015 Conference on Neural Information Processing Systems(NIPS), Montreal, 2015, 1693-1701.

[17] A. See, P. J. Liu, C. D. Manning. Get to The Point: Summarization with Pointer-Generator Networks[C]. Proceedings of the 55th Conference on Association for Computational Linguistics (ACL), Vancouver, 2017, 1073-1083.

[18] R. Paulus, C. Xiong, R. Socher. A Deep Reinforced Model for Abstractive Summarization[OL]. https://arxiv.org/pdf/1705.04304.pdf.

[19] J. Devlin, M. W. Chang, K. Lee, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]. Proceedings of the 2019 Conference on the North American Chapter of the Association for Computational Linguistics, Minneapolis, 2019, 4171-4186.

[20] U. Khandelwal, K. Clark, D. Jurafsky, et al. Sample Efficient Text Summarization Using a Single Pre-Trained Transformer[OL]. https://arxiv.org/abs/1905.08836.

[21] S. Subramanian, R. Li, J. Pilault, et al. On Extractive and Abstractive Neural Document Summarization with Transformer Language Models[OL]. https://arxiv.org/abs/1909.03186v1.

[22] 苏海菊, 王永成. 中文科技文献文摘的自动编写[J]. 情报学报, 1989, 8(06): 433-439.

[23] 程园, 吾守尔·斯拉木, 买买提依明·哈斯木. 基于综合的句子特征的文本自动摘要[J]. 计算机科学, 2015, 42(04): 226-229.

[24] 郭艳卿, 赵锐, 孔祥维, 等. 基于事件要素加权的新闻摘要提取方法[J]. 计算机科学, 2016, 43(01): 237-241.

[25] B. Hu, Q. C. Chen, F. Z. Zhu. LCSTS: A Large Scale Chinese Short Text Summarization Dataset[C]. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, 2015, 1967-1972.

[26] J. Gu, Z. D. Lu, H. Li, et al. Incorporating Copying Mechanism in Sequence-to-Sequence Learning[C]. Proceedings of the 54th Conference on ACL, Berlin, 2016, 1631-1640.

[27] Ayana, S. Shen, Y. Zhao, et al. Neural headline generation with sentence-wise optimization[OL]. https://arxiv.org/abs/1604.01904

[28] 周才东, 曾碧卿, 王盛玉, 等. 结合注意力与卷积神经网络的中文摘要研究[J]. 计算机工程与应用, 2019, 55(08):132-137.

[29] Dataset of NLPCC2017task#3[OL]. http://tcci.ccf.org.cn/conference/2017/taskdata.php.

[30] L. Hou, P. Hu, C Bei. Abstractive Document Summarization via Neural Model with Joint Attention[C]. Proceedings of the 2017 Conference on Natural Language Processing and Chinese Computing, Dalian, 2017, 329-338.

[31] R Sennrich, B. Haddow, A. Birch. Neural Machine Translation of Rare Words with Subword Units[C]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, 2016, 1715-1725.

[32] P. Gage. A New Algorithm for Data Compression[J]. C Users Journal, 1994, 12(2): 23-38.

[33] 侯丽微, 胡珀, 曹雯琳. 主题关键词信息融合的中文生成式自动摘要研究[J]. 自动化学报, 2019, 45(03): 530-539.

[34] A. Radford, J. Wu, R. Child, et al. Language Models are Unsupervised Multitask Learners [OL]. https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf

[35] A. Dasgupta, P. Drineas, B. Harb, et al. Feature selection methods for text classification[C].Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, San Jose, 2007, 230-239.

[36] S. Günnemann, H. Kremer, D. Lenhard, et al. Subspace clustering for indexing high dimensional data: a main memory index based on local reductions and individual multi-representations[C]. Proceedings of the 14th International Conference on Extending Database Technology(EDBT), Uppsala, 2011, 237-248.

[37] A. P. Vries, N. Mamoulis, N. Nes, et al. Efficient k-NN search on vertically decomposed data[C]. Proceedings of the 2002 ACM SIGMOD international conference on Management of data. Madison, 2002, 322-333.

[38] Y. C. Liaw, M. L. Leou, C. M. Wu. Fast exact k nearest neighbors search using an orthogonal search tree[J]. Pattern Recognition, 2010, 43(6): 2351-2358.

[39] M. M. Hassan, C. M. Text Categorization using association rule based decision tree[C]. In proceedings of 6th International Conference on Computer and Information Technology, Korea, 2003, 453-456.

[40] Y. Kim. Convolutional neural networks for sentence classification[C]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, 2014, 1746-1751.

[41] N. Kalchbrenner, E. Grefenstette, P. Blunsom. A Convolutional Neural Network for Modelling Sentences[C]. Proceedings of the 2014 Conference on the Association for Computational Linguistics. Baltimore, 2014, 655-665.

[42] S. Lai, L. Xu, K. Liu, et al. Recurrent convolutional neural networks for text classification[C]. Twenty-ninth AAAI conference on artificial intelligence. Austin, 2015, 2267-2273.

[43] A. Joulin, E. Grave, P. Bojanowski, et al. Bag of tricks for efficient text classification[C]. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. Valencia, 2017, 427-431.

[44] Z. Yang, D. Yang, C. Dyer, et al. Hierarchical attention networks for document classification[C]. Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. San Diego, 2016, 1480-1489.

[45] S. Gao, A. Ramanathan, G. Tourassi. Hierarchical convolutional attention networks for text classification[C]. Proceedings of The Third Workshop on Representation Learning for NLP. Melbourne, 2018, 11-23.

[46] X Zhou, X. Wan, J. Xiao. Attention-based LSTM network for cross-lingual sentiment classification[C]. Proceedings of the 2016 conference on empirical methods in natural language processing. Austin, 2016, 247-256.

[47] Y. Zhou, C. Li, B. Xu, et al. Hierarchical Hybrid Attention Networks for Chinese Conversation Topic Classification[C]. International Conference on Neural Information Processing. Guangzhou, 2017, 540-550.

[48] 李荣陆, 胡运发. 基于密度的 kNN 文本分类器训练样本裁剪方法[J]. 计算机研究与发展, 2004, 41(4):539-545.

[49] 刘赫, 刘大有, 裴志利, 等. 一种基于特征重要度的文本分类特征加权方法[J]. 计算机研究与发展, 2009, 46(10): 1693-1703.

[50] 张杰慧, 何中市, 王健, 等. 基于自适应蚁群算法的组合式特征选择算法[J]. 系统仿真学报, 2009, 21(6): 1605-1608.

[51] 李楠, 杨彬彬. 决策树 ID3 分类算法在文本分类中的应用研究[J]. 大连大学学报, 030(6):68-71.

[52] 单丽莉, 刘秉权, 孙承杰. 文本分类中特征选择方法的比较与改进[J]. 哈尔滨工业大学学报, 2011, 43(3): 319-324.

[53] 蒋胜利. 高维数据的特征选择与特征提取研究[D]. 西安：西安电子科技大学, 2011, 1-119.

[54] 李静, 杨小帆, 孙启干. 面向 Web 信息检索的虚核文本分类算法[J]. 计算机工程, 2012, 38(10): 182-184+ 187.

[55] 赵明, 杜会芳, 董翠翠, 等. 基于 word2vec 和 LSTM 的饮食健康文本分类研究[J]. 农业机械学报, 2017, 48(10): 202-208.

[56] 梁斌, 刘全, 徐进, 等. 基于多注意力卷积神经网络的特定目标情感分析[J]. 计算机研究与发展, 2017, 54(8): 1724-1735.

[57] 卢玲, 杨武, 王远伦, 等. 结合注意力机制的长文本分类方法[J]. 计算机应用, 2018, 38(5): 1272-1277.

[58] 陈珂, 梁斌, 柯文德, 等. 基于多通道卷积神经网络的中文微博情感分析[J]. 计算机研究与发展, 2018, 55(5): 945-957.

[59] 杜雨萌, 张伟男, 刘挺. 基于主题增强卷积神经网络的用户兴趣识别[J]. 计算机研究与发展, 2018, 55(1): 188-197.

[60] 刘腾飞, 于双元, 张洪涛, 等. 基于循环和卷积神经网络的文本分类研究[J]. 软件, 2018 (2018 年 01): 64-69.

[61] 王根生, 黄学坚. 基于 Word2vec 和改进型 TF-IDF 的卷积神经网络文本分类模型[J]. 小型微型计算机系统, 2019, 40(5): 1120-1126.

[62] 陈志, 郭武. 不平衡训练数据下的基于深度学习的文本分类[J]. 小型微型计算机系统, 2020, 41(01): 1-5.

[63] 段丹丹, 唐加山, 温勇, 等. 基于 BERT 的中文短文本分类算法的研究[OL]. https://doi.org/10.19678/j.issn.1000-3428.0056222.

[64] Z. Tu, Z. Lu, Y. Liu, et al. Modeling coverage for neural machine translation[C]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, 2016, 1453-1461.

[65] W. Y. Zeng, W. J. Luo, S. Fidler, et al. Efficient summarization with read-again and copy mechanism[OL]. https://arxiv.org/abs/1611.03382.

[66] X. P. Jiang, P. Hu, L. W. Hou, et al. Improving pointer-generator network with keywords information for Chinese abstractive summarization[C]. Proceedings of the 7th International Conference on Natural Language Processing and Chinese Computing, Hohhot, 2018, 464-474.

[67] L. Page, S. Brin. The PageRank Citation Ranking: Bringing Order to the Web[OL]. http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf.

[68] A. Valerie, H. Suzanne. Teaching students to summarize[J]. Educational leadership, 1988, 46(4): 26–28.

[69] P. Anderson, X. He, C. Buehler, et al. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering[C]. Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018, 6077-6086.

[70] 方丹. 网络警情的信息提取与分析的关键技术研究与实现[D]. 电子科技大学, 2019, 1-85.

[71] 徐馨韬, 柴小丽, 谢彬, 等.基于改进 TextRank 算法的中文文本摘要提取[J]. 计算机工程, 2019, 45(03): 273-277.

攻硕期间的研究成果

获奖情况：

[1] 2017 年 10 月，研究生第一学年二等奖学金

[2] 2018 年 10 月，研究生第二学年二等奖学金

[3] 2019 年 10 月，研究生第三学年三等奖学金