法律问题关键词抽取系统

摘要

法律问题关键词抽取是当前关键词抽取技术在我国法律范畴内的应用，法律问题与人们日常生活以及社会规范制度息息相关。目前我国的法律文件及条款在我国人口众多、社会情况复杂等诸多原因下不断完善且数据量庞大，且伴随着高速网络信息和大数据时代的到来，绝大部分的人们由于专业限制通过使用各种在线法律问答系统来咨询自身所需的法律咨询或援助。如何从法律问答库的海量数据中抽取出人们需要的、准确的数据已经是当前研究的重点。关键词抽取是法律信息挖掘最基本的任务，准确的关键词能够快速引导人们关于法律问题的参考方向，能够告知人们某个法律问题所属的具体法律条款范畴。由于当前关键词抽取技术大多在长文本中基于统计学的方法，在法律问答（短文本）的关键词抽取效果不好。

因此本文主要通过构建一个基于循环神经网络(RNN)的序列到序列(seq2seq)模型从某一简短的法律问题中生成关键词。首先从法律问答网站上利用爬虫技术收集大量法律问题以及已经人为对应标记好的关键词，并对（问题-关键词）语料库做预处理。对语料库进行中文分词之后，利用word2vec模型训练生成词向量。然后使用已经构建好的seq2seq模型训练，从而能够让训练好的模型预测生成准确的关键词。

关键词：关键词抽取；神经网络；序列模型；word2vec；机器学习

**LEGAL ISSUES KEYWORDS EXTRACTION SYSTEM**

**ABSTRACT**

Keywords extraction of legal issues is the application of current keyword extraction technology in China's legal scope,legal issues are closely related to people's daily lives and social norms.At present, China's legal documents and terms are constantly improving and the amount of data is huge due to many reasons such as China's large population and complex social conditions.The vast majority of people seek their own legal advice or assistance through the use of various online legal question and answer systems due to professional restrictions.How to extract the accurate data that people need from the large amount of data of the legal question and answer library is the focus of current research.Since most of the current keyword extraction techniques are based on statistical methods in long texts, the keyword extraction in legal questions and answers (short text) is not good effective.

Therefore, this paper mainly generates keywords from a short legal issue by constructing a sequence-to-sequence (seq2seq) model based on the Recurrent neural network (RNN).First, from the legal Q&A website, we use Crawler technology to collect a large number of legal questions and keywords that have been artificially labeled, and pre-process the (question-keyword) corpus.After the Chinese word segmentation of the corpus, the word2vec model is used to generate the word vector. Then use the already built seq2seq model to train, so that the trained model can predicts accurate keywords.

**Keywords:** Neural Networks; Sequence model; Word2vec; Machine Learning

目 录

[1 绪论 1](#_Toc3088)

[1.1 课题背景及研究的目的 1](#_Toc6798)

[1.2 国内外研究现状及发展趋势 2](#_Toc16157)

[1.2.1 国内外研究现状 2](#_Toc7046)

[1.2.2 发展趋势 3](#_Toc14715)

[1.3 课题主要研究内容 4](#_Toc20927)

[2 开发工具及相关技术 5](#_Toc20347)

[2.1 系统开发工具 5](#_Toc27613)

[2.1.1 Python3.6 5](#_Toc23142)

[2.1.2 Pycharm 5](#_Toc30324)

[2.1.3 NVIDIA GEFORCE GTX 850M 5](#_Toc4925)

[2.2 中文分词 5](#_Toc12852)

[2.2.1 基于字符匹配的分词方法 6](#_Toc23752)

[2.2.2 基于理解的分词方法 6](#_Toc20568)

[2.2.3 基于统计的分词方法 7](#_Toc16052)

[2.2.4 jieba分词 8](#_Toc28587)

[2.3 词嵌入技术 9](#_Toc8980)

[2.3.1 word2vec技术 9](#_Toc18083)

[2.3.2 CBOW模型和Skip-gram模型 9](#_Toc12421)

[2.4 Pytorch机器学习框架 11](#_Toc2913)

[3 带注意力机制的seq2seq模型 12](#_Toc11963)

[3.1 循环神经网络 12](#_Toc28817)

[3.1.1 神经网络的概念 12](#_Toc4174)

[3.1.2 RNN模型结构及原理 14](#_Toc1673)

[3.2 seq2seq模型原理 15](#_Toc21490)

[3.3 attention机制 16](#_Toc2580)

[4 系统分析 19](#_Toc16524)

[4.1 数据需求分析 19](#_Toc14017)

[4.2 功能需求分析 19](#_Toc23223)

[4.3 性能需求分析 20](#_Toc8389)

[5 系统实现 21](#_Toc10172)

[5.1 语料收集和预处理 21](#_Toc31392)

[5.1.1 数据爬虫 21](#_Toc12558)

[5.1.2 数据预处理 22](#_Toc3361)

[5.2 中文分词 22](#_Toc5312)

[5.3 词向量生成 22](#_Toc7310)

[5.4 搭建基于强化学习的seq2seq模型 22](#_Toc4753)

[5.4.1 编码器-解码器 22](#_Toc3058)

[5.4.2 NLLLoss损失函数 22](#_Toc17029)

[5.5 训练模型 22](#_Toc22489)

[5.5.1 带监督的训练 22](#_Toc18727)

[5.5.2 使用GPU加速训练 22](#_Toc3083)

[5.6 系统运行与测试评估 22](#_Toc16099)

[6 系统测试 23](#_Toc13317)

[7 总结 24](#_Toc27412)

[参考文献 25](#_Toc1080)

[致谢 26](#_Toc14532)

[附录:程序主要源代码 27](#_Toc10851)

1 绪论

1.1 课题背景及研究的目的

在互联网大数据时代背景下，互联网使用人数以及各种终端设备如智能手机、平板、笔记本爆发式增长。信息时代为人们带来便捷的同时，网络上庞大且冗杂的数据也让人们常常需要花费许多精力去筛选得到想要的信息。百度、Google等搜索引擎在一定程度上协助人们在如此庞复的网络数据中找到需要的信息，然而搜索引擎只支持使用关键词进行检索，且在日益增长的数据量情况下对关键词精准度的要求也越来越高。关键词是简短且具有高度总结性的文本，需要用户根据自身知识来概括提取。但在法律问题方面，由于用户专业限制，无法保证依靠人工提取关键词的精准性，从而也影响了搜索引擎返回信息结果的准确性。

我国坚持依法治国的基本方略，保证社会治安稳定、社会文明进步以及国家长治久安。要求人民和执法部门有法必依、违法必究、执法必严，人们的法律意识和知识水平也日益提高。随着我国法律制度体系越来越完善，法律条款及文件的数量越来越大且复杂，非法律专业人士由于专业限制无法定位具体的法律范畴。法律社区问答网站（如：www.51wf.com）的出现给人们提供了一个很好的专业法律咨询的平台，用户在网站上提交一段较为简短、口语化的法律问题，会有法律专业人士提供非常专业的回答，并将该问题打上标签作为其关键词，能够迅速定位用户所咨询问题所属的法律专业范畴。本课题研究的目的是通过收集上述法律社区问答网站上的大量法律问题-关键词数据，运用自然语言处理、机器学习等技术训练，得到一个可准确挖掘出法律问题中包含的关键词的模型。该模型能够根据用户输入的法律问题，自动生成关键词，帮助用户定位问题的范畴并指引用户搜索的方向，能够大大提高用户获取需要的法律信息的效率。

1.2 国内外研究现状及发展趋势

### 1.2.1 国内外研究现状

关键词是一段简短的总结性内容，用来表达文本的主要语义，高质量的关键词可以促进对文本内容的理解，组织和访问。因此，国外许多研究都集中在从文本内容中自动提取关键短语的方法，它已广泛应用于许多应用。如Jones和Staveley在1999年提出信息检索、Hulth和Megyesi在2006年提出的文本分类、Berend提出的意见挖掘等应用。大多数现有的关键词提取算法通过两个步骤解决了这个问题，第一步是获取关键词候选列表，研究人员试图使用具有某些词性模式的n-gram或名词短语来识别潜在的候选关键词，如Medelyan从维基百科等重要语料库中提取n-gram，并根据预先建立的规则提取名词或短语。第二步是通过有监督或无监督的机器学习方法和一系列人为定义的特征将候选关键词对文本内容的重要性进行排名。在带监督的机器学习中，关键词提取的任务可以归为文本分类问题，如Frank使用朴素贝叶斯训练分类器；在无监督的机器学习中，Mihalcea等人提出计算候选关键短语之间的相关性。

上述关键词提取方法主要存在两个主要缺点：首先，这些方法只能提取源文本中出现过的关键词；它们无法预测具有略微不同顺序的有意义的关键词或使用同义词的关键词。然而，法律问答系统通常根据其语义来分配关键词，而不是根据文本内容。由于法律问题是由非专业用户口语化提出的，关键词往往不包含在问题中，所以通过上述的方法提取效果不佳，这进一步促使开发更强大的关键词预测模型。其次，在对候选关键词进行排名时，以前的方法通常采用机器学习算法，如TF-IDF和PageRank。然而，这些特征仅基于词语出现和共现的统计来检测文档中每个词语的重要性，并且不能揭示作为文本内容基础的完整语义。为了解决短文本中关键词难以使用统计方法抽取的问题，诸多学者已着手研究从短文本中生成关键词的技术，如Zhang等人提出了一种用于短文本提取关键词的递归神经网络模型。但是，这些模型仍然不能抽取原始文本中未出现的关键词。

在国内，法律问题关键词自动提取也有不少研究。由于中文不像英文那样有明显的分词界限，且中文是世界上语法最复杂困难的语言之一，这在一定程度上增加了中文关键词提取的难度。在中文关键词提取之前，需要对语料进行分词处理，目前许多研究人员已经提出若干优秀的中文分词工具，如HanLP、jieba分词、FudanNLP等。在基于语义的关键词提取方法中，王立霞等提出使用《同义词词林》计算每个词语的语义相关程度,构建一个语义相似度网络,利用中介密度来衡量每个词语的重要性并将词语的语义特征整合到关键词提取的过程中,效果较好。

### 1.2.2 发展趋势

由于我国疆域辽阔人口众多，社会情况和地方文化复杂，我国的法律文献及条款众多并且将来还会不断的修订和完善。因此整个法律体系所涉及的法律问题类型分类以及各种法律问题检索的复杂度会越来越高。当人们碰到法律问题时往往只能够做出口语化的描述，并不能准确的定位某个法律问题所属的法律专业范畴。法律社区问答网站的出现使人们咨询法律的途径越来越便捷，但是同时由于网络信息数量巨大，且大部分法律社区问答网站需要通过专业人士人为解答、人为标记关键词，网站无法立即反馈准确关键词信息给用户。当前的法律问题关键词提取技术和法律问题分类还不是特别完善，这就有可能产生用户被误导，混淆法律概念的后果，因此要想真正满足人们和社会的需求，法律问题关键词提取方面还需进一步探索，主要体现在：

（一）准确性：法律问题与个人、公司或部门、政府等单位的利益以及社会制度密切相关，当涉及到这些内容时，法律条款和文件起到了决定性的作用。这就要求法律问题关键词抽取系统能够给出精准的关键词来定位问题所属法律范畴。

（二）易用性：法律问题一般是由非专业人士提出的，由于我国法律文件包含的专业词汇数量巨大，普通人无法准确使用专业词汇描述某一法律问题，描述内容一般偏口语化。再者中文本身语法复杂多变且不断有新的网络流行词汇诞生，所以如何从用户口语化描述的问题中提取专业法律词汇关键词将成为今后研究的重点。

（三）完整性：社会不断的发展，我国法律文献也随之不断修订和完善。法律关键词抽取模型必须参照当前最新最完整的法律文献库，这就要求研究人员以及专业法律机构共同努力收集全面的法律语料，保证模型能够涵盖所有用户需要的法律问题关键词。

1.3 课题主要研究内容

本课题主要研究一种基于序列到序列的法律问题关键词抽取系统，系统主要分为语料收集和预处理、中文分词、词向量训练、搭建基于强化学习的序列到序列模型、训练模型等模块。

语料收集和预处理：从法律社区问答网站（www.51wf.com）收集大量的法律问题语料，并对数据进行特定格式化以及清理，提供给之后的模块训练。

中文分词：采用综合了基于字符串匹配的算法和基于统计的算法的jieba分词工具进行法律语料的分词工作。

词向量训练：使用Word2vec的CBOW模型，对分词后的法律语料进行训练，生成所有词汇的向量模型。

搭建基于强化学习的seq2seq模型：构建基于seq2seq的RNN模型，将处理完的预料作为输入输出序列，在其中引入强化学习用来训练语料库。

训练模型：在NVIDIA GEFORCE GTX 850M上使用Pytorch调用其CUDA架构进行GPU加速训练，生成一个能够预测关键词的模型。

2 开发工具及相关技术

2.1 系统开发工具

### 2.1.1 Python3.6

Python是一种解释性语言，它的易学性、可移植性以及拥有丰富强大的标准库等特点，使其成为在进行机器学习、神经网络等开发时的首选编程语言。目前许多优秀的开源机器学习框架都是基于Python语言编写的。

### 2.1.2 Pycharm

Pycharm是一款强大的Python IDE，它拥有完整的集成开发环境，如图形化界面、代码编辑器、解释器、调试器，在编程时能够大大提高开发效率以及调试效率。

### 2.1.3 NVIDIA GEFORCE GTX 850M

GeForce GTX 850M 是伟英达（NVIDIA ）公司旗下出品的一款中端笔记本级别显卡。该款显卡提供CUDA，GPU Boost 2.0以及3D Vision等技术支持，其中CUDA 是 NVIDIA 发明的一种并行计算平台和编程模型。它通过利用图形处理器 (GPU) 的处理能力，可大幅提升计算性能。由于本课题所训练的语料数据庞大，使用CPU训练效率低，故在机器学习过程中使用GPU来提高训练速度。

2.2 中文分词

中文分词，即把一段中文语句按照句子本身语义分割为以词语为基本单元的词序列。众所周知，在英文句子语法中，每个单词之间都有空格作为天然的分隔符，然而中文句子中的词、字并没有明显的分隔符，且基于中文语法复杂的特点，同一中文句子中的词语可能有存在歧义的不同分割法。法律问题往往包含许多专业词汇的简称，进一步加大了词语划分的难度，故清晰准确的中文分词是本系统NLP（Natural Language Processing，自然语言处理）任务的首要指标。

### 2.2.1 基于字符匹配的分词方法

基于字符匹配的分词方法，是按照特定的策略将待分词的中文语句在一个已建立好的“充分大的”词典中的词进行匹配。若匹配到一个词条，说明词典成功识别了该词，就将其分割。故该方法也被称为基于词典的分词方法或机械分词法，这种匹配的算法按照扫描的方式又可以分成以下四种匹配法：

（一）正向最大匹配算法

正向最大匹配算法，指的是利用类似于滑动窗口概念从左到右滑动依次匹配，该方法首先给定一个词典中最大词条的长度作为滑动窗口的长度L，根据滑动窗口的位置取出对应的L个中文字符进行匹配。若未匹配，则L-1并重复上述步骤，直到匹配成功为止。

（二）逆向最大匹配算法

逆向最大匹配算法，该算法是正向最大匹配算法的逆思维，即从右往左对中文语句进行最大匹配。根据网络上大量实验表明逆向最大匹配算法效果要优于正向最大匹配算法。

（三）双向最大匹配算法

大部分情况下，前向匹配和后向匹配得到的结果是相同的，但是也有不同的情况。双向最大匹配就是同时进行前行和后向最大匹配，然后分析两种的结果。如果两种结果一致，则认为不存在歧义现象；如果不一致，则需要定位到歧义字段处理。优点是提高了分词的准确率，消除了部分歧义现象。缺点是算法执行要做双向扫描，时间复杂度会有所增加。

（四）最少切分算法

不同于前述三种算法，最少分切算法使用动态规划的思想，将分词任务划分成局部问题，使每一句中切出的词数最小。该算法需要遍历中文语句所有可能的切分路径，这也导致了计算量过大的缺点。

### 2.2.2 基于理解的分词方法

基于理解的分词方法，其核心思想就是让机器模拟人类大脑的思维方式来理解句子、分割词语。通过对中文语句的语义、句法分析后进行分词，达到能够处理分词过程中歧义部分的效果，故这种方法也常称为基于语义的分词法。

该方法需要建立一个足够大的词典并且词典中包含所有词语与之对应的所有语义信息，然后根据某种匹配算法分割出子串S，若S与词典中某个词W相匹配，则取出W的语义信息并结合句子对S进行语义分析。若符合该句子的语义信息，则认为分析正确，接着分割句子剩余部分。

这种方法的缺点在于，需要构建一个庞大的语言知识和语义信息库。由于中文语言的复杂性、歧义性，在法律问答等口语化情境下，难以使用标准化句法结构来分析口语化句型。

### 2.2.3 基于统计的分词方法

基于统计的分词方法的核心是把每个词看作是由单个汉字组成，如果相连的汉字在不同文本中出现的次数越多，就证明这段相连的字很有可能就是一个词。一般分为两个步骤，首先构建统计语言模型（如n-gram），第二步对句子进行单词划分，然后对划分结果做概率计算，获取概率最大的分词方式。这里就用到了统计学中的算法，如隐马尔科夫模型（HMM）。

设有一段长度为i的中文语句S：

 (2-1)

S的所有可能的切分路径：

 (2-2)

目的是找出使得条件概率最大的切分路径:

 (2-3)

根据Bayes公式：

 (2-4)

计算得出：

 (2-5)

在n-gram模型中，当n=2表示当前词仅与前一个词有关，记作bi-gram。二 元文法模型被称为一阶马尔科夫，则有：

 (2-6)

这种方法的优点在于不用构建一个巨大的词典，不受法律等特殊范畴的词汇限制，并且具有分析语义信息，处理歧义内容的功能。缺点在于统计过程中需要一个庞大的语料库作为基础，并且容易抽取出对句子语义影响不大但共现概率很大的一些停用词，如：“同一”，“白白”。

2.2.4 jieba分词

jieba是目前最好的Python中文分词组件，它综合了基于字符串匹配的算法和基于统计的算法的优点。它包含一个很大很全的词典，即dict.txt文件，实现了HMM隐马尔科夫模型。并且支持添加自定义词典以及调整词典的功能，能够有效解决法律专业词汇简称无法识别的问题。

它支持三种分词模式：

（1）精确模式：尝试最精确地分割句子，适合分析文本内容。

（2）全模式：扫描句子中所有可以用于措辞的词语, 速度快。

（3）搜索引擎模式：基于精确模式，再次分割长词，提高召回，适用于搜索引擎。

综合上一小节对三种分词类别各自优缺点的分析，以及jieba库强大的功能组件且比较适合法律问题具体的句法情境，本系统使用jieba库进行法律语料数据的中文分词工作。

## 2.3 词嵌入技术

One-Hot编码是词向量的一种简单、基本的表示方法，其思想是使用N维向量来对N个词进行编码，每个词的向量值唯一的。并且在任意时候，所有向量只有一位为1，其余位都是0。例如，假设“盗窃”和“抢劫”这两个词对应词典中的索引分别为2和4，则它们的One-Hot编码表示为：

盗窃：

抢劫：

使用这种编码的优点在于简洁易懂，但是假设在有2万个词汇构成的词典中，每个词向量的维度将达到2万，尽管其中只有一位是有效位。这种情况下有可能造成词向量过于稀疏和维度爆炸的后果。词的分布式表示（Dristributed representation）思想是通过训练，将每个词都映射到一个较短的词向量上来，通过密集低维的向量有效解决维度爆炸的问题，这个过程又称为词嵌入（Word embedding）。

### 2.3.1 word2vec技术

word2vec是Google在2013年开源发布的一种计算计算词向量的工具，它能够对数百万级字典和数亿个数据集进行高效训练。该工具训练得到的词向量可以非常好地衡量单词和单词之间的相似度。word2vec包括两种模型：Skip-Gram模型和CBOW模型。

### 2.3.2 CBOW模型和Skip-gram模型

（一）CBOW模型

CBOW（Continuous Bag-of-Word Model），该模型的特点在于输入词序列中某个目标词X的上下文单词，输出对X的预测。因为它用到了连续的词语，且忽略这些词之间的顺序，所以该模型又称作连续词袋模型，其模型结构图2.1所示：

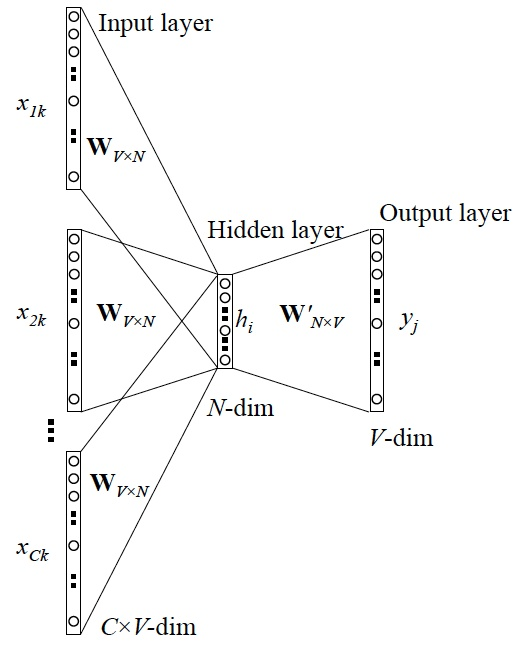


图 2.1 CBOW 模型网络结构图

模型中，Input layer(输入层)内容为上下文所有单词的One-Hot编码，Hidden layer(隐含层)内容为输入层所有向量相加和的平均值：

 (2-7)

传统模型的输出层采用softmax函数计算单词分布概率，计算复杂度高，耗时长。所以CBOW使用Hierarchy Softmax来提高计算效率，该方法思路是在模型输出层建立一棵哈夫曼树。根据哈夫曼树的特点，词典中每个词代表一个叶子节点，从而计算量就从单词个数降到路径分支条数，计算复杂度从降到。

（二）Skip-gram模型

Skip-gram模型则恰好与CBOW模型相反，它是根据输入的目标词X去预测其的上下文单词，其模型结构图2.2所示：

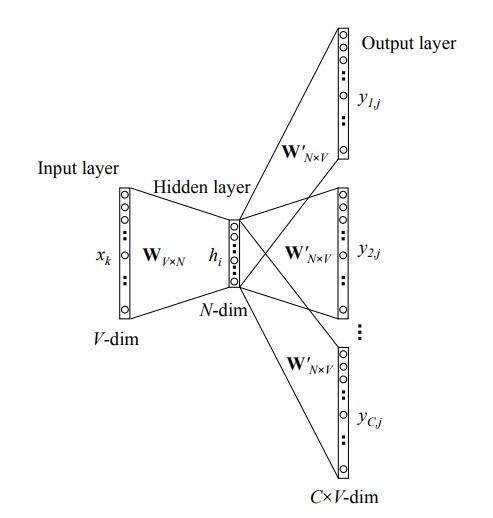


图 2.2 Skip-gram模型网络结构图

模型中，Input layer(输入层)内容为目标单词X的One-Hot编码，Hidden layer(隐含层)内容为目标单词X对应权重矩阵W的词向量：

 (2-8)

基于本课题的特点，法律问题语料库结构为输入多个词构成的序列，输出一个或多个关键词。简单对比两个模型可以得知CBOW模型更加适合法律问题关键词抽取任务。

2.4 Pytorch机器学习框架

Pytorch是Facebook人工智能研究院（FAIR）团队于2017年开源的一款机器学习框架，它具有简洁灵活、优雅易用、高效快速等优势。其结构为tensor→autograd→nn.Module 三个由低到高的抽象层次，分别代表张量、自动求导和神经网络模块；速度方面，在网上许多评估表明Pytorch的速度超过TensorFlow、Keras等框架。易用性方面，Pytorch代码量少、功能更加直观，更加符合人类思维。目前一些著名的公司如twitter、Nvidia都在使用该框架。

3 带注意力机制的seq2seq模型

一段法律问题文本根据人类思维可以分为法律范畴决定词和语义铺垫词，例如有这样一则法律问题：“我在工作期间不小心发生了交通事故，该由谁赔偿？”，以人类的注意力思维可以得出“交通事故”、“赔偿”是这此问题所属法律范畴的决定性词语，而“不小心”、“发生”等词对该问题语义影响不大。为了使机器能够模拟人类的注意力思维来提高学习效率，本文提出了一种带注意力（Attention）机制的seq2seq模型。

3.1 循环神经网络

3.1.1 神经网络的概念

神经网络（Neural Network）是指使用计算机技术来模拟人类大脑神经元网络，将具有适应性的基本神经单元构成广泛互联的网络，它能够使计算机具有人脑神经结构特征，来模拟生物神经网络对真实物理世界的理解与交互。

（一）神经网络的组成

（1）神经元模型

在生物神经网络中，每个神经元都与其他神经元相连。当某个神经元被激发时，它会向连接的其他神经元反馈信息，从而改变这些神经元的状态。若某个神经元的状态超过一个特定的阈值，那么它将被激活，使得该神经元状态置为兴奋并告知大脑。将这一模型抽象出来，如图3.1所示：

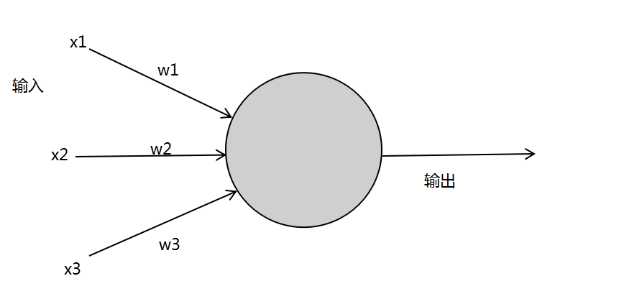


图3.1 神经元模型

（2）权重和阈值

权重（weight），在图3.1中，为输入值，为对应的权重值。输入和权重可以分别理解为现实世界中产生某个结果的决定性因素，和每个决定性因素对结果的重要性。

阈值又称为偏置（bias），如上面描述，它是神经元被激活与否的分界线，它的大小更够直观的反映神经元产生正（负）激励的难易度。阈值越高表明神经元越难被激活，反之阈值越低则神经元越容易被激活。

（3）激活函数

激活函数（Activation functions），如图3.2所示神经元中，输入的序列经过加上对应的权重求和后，还输出层之前还经过了一个激活函数的运算，目的是为了增加该神经网络模型 的非线性。使用激活函数的必要性体现在当一个神经网络具有多层结构时，输入和输出都是矩阵相乘的线性组合。常用的激活函数有Sigmoid、Tanh、ReLU等。

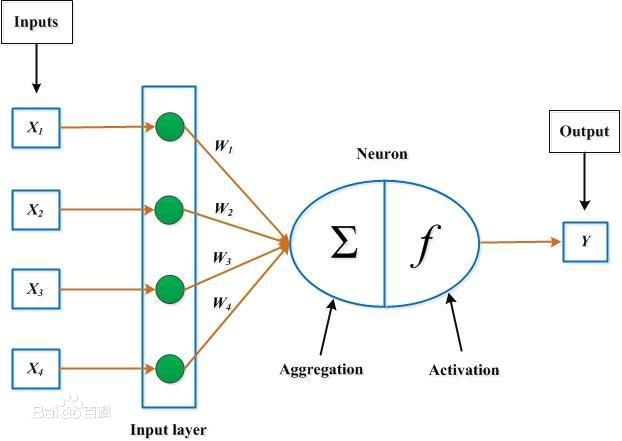


图3.2 神经网络激活函数

3.1.2 RNN模型结构及原理

（一）单层神经网络

通过前述对神经网络的结构‌‌ 分析，我们容易得到如图3.3所示的基本单层神经网络结构示意图，其中输入为,箭头表示经过激活函数与变换得到输出。

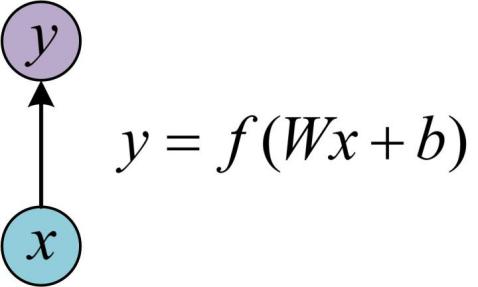


图3.3 单层神经网络结构

(二)RNN

在上述的单层网络模型中，只能单独处理每个输入且每个输入的前后顺序和时间顺序完全无关。在自然语言处理过程中，单独地分析一句话中的每个词是行不通的，我们需要理解这些词连接起来的序列。序列数据用单层网络模型处理效果不佳，因此使用RNN（Recurrent Neural Network，循环神经网络）。RNN模型中引入了hidden state（隐含状态）的概念，其模型结构图如图3.4所示：

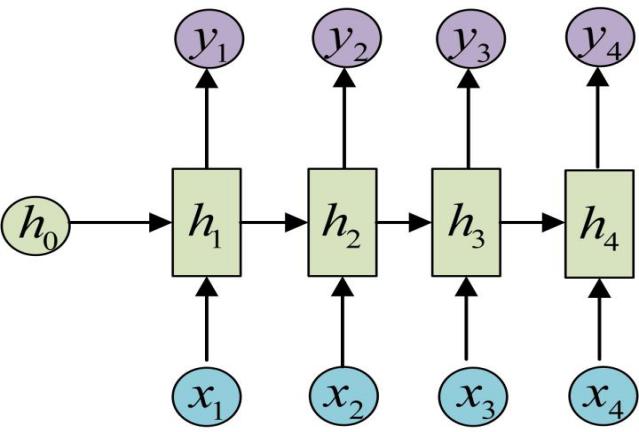


图3.4 RNN结构示意图

其中，为输入序列，为对应输入的隐含状态，为对应输出，为初始状态，箭头表示对向量做了一次变换。隐状态，为计算的共享参数，以此类推。输出,以此类推，可以看出每个输出的结果与当前输入和上一次输入都有关系，它能够将整个序列的词语相关联。

以上为RNN最经典的一种结构（N vs N），也就是说输入序列与输出序列是等长的，这就导致了该结构在自然语言处理方面的局限性。因此这里介绍RNN的另外两种结构：

（1）N vs 1

有时候，我们需要输入一个序列，输出的是单个值而不是序列，建模时可以仅仅在最后一个隐状态进行变换输出，如图3.5所示：

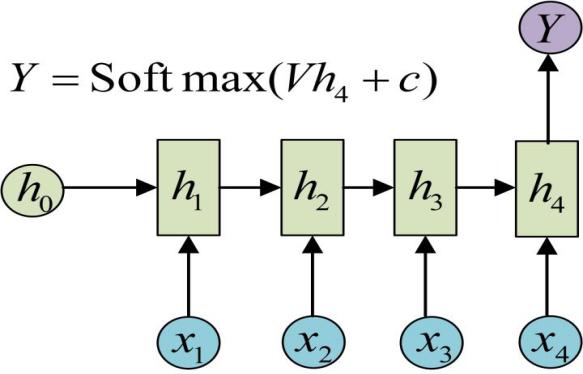


图3.5 N vs 1结构

该结构在在处理序列分类如判断句子感情分类等应用中使用较为广泛。

（2）1 vs N

与N vs 1结构恰好相反，1 vs N结构接受单个输入，输出则为一个序列，建模时可以将单个的输入分别作为每个阶段的输入，如图3.6所示：

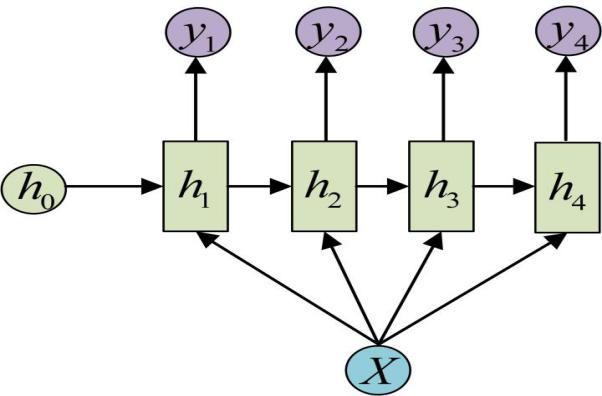


图3.6 1 vs N结构

该结构能够处理的问题有如利用情感类别生成评论数据等。

3.2 seq2seq模型原理

前面介绍了RNN常见的3种结构：N vs N、N vs 1、1 vs N。在如图3.7所示法律问题中，法律问题的词序列作为输入，关键词标签作为输出。

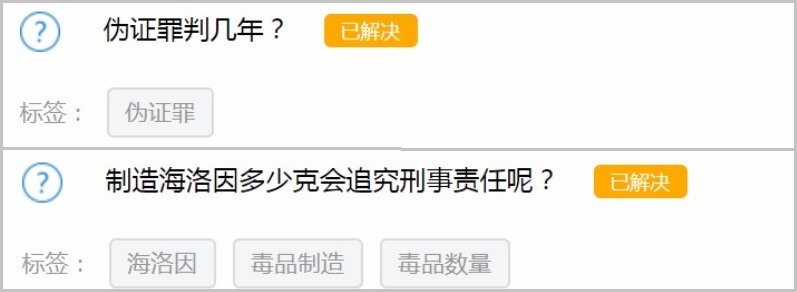


图3.7 法律问题-关键词结构

可以看出，输入的序列对应的输出可以对应单个关键词，也可以是一个关键词序列，所以在处理法律问题关键词方面，上述的三种结构都不能满足要求。因此介绍一个RNN重要的变种：N vs M结构，也称为Encoder-Decoder（编码器-解码器）模型或者seq2seq模型。如图3.8所示：

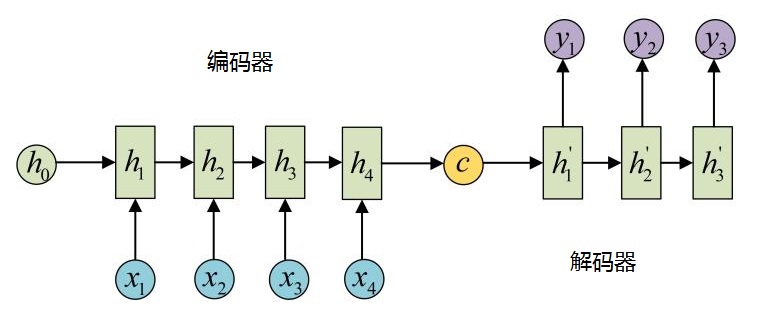


图3.8 N vs M结构

首先，编码器将输入的序列编码成一个上下文语义向量 ,然后再由解码器进行解码，在解码的过程中当前时刻不断以上一时刻的输出作为参数，循环解码直至输出停止符。

3.3 attention机制

在Encoder-Decoder‌‌‌模型中，Encoder‌将‌‌‌‌‌‌输入序列 编码成 一个统一 的上下文语义 向量,然后交给Decoder进行.解码。由于中储存了原始序列的所有语义信息，若是在句子字数比较多的情况下，一些对语义影响不大的词语也被压入中，导致精度和性能的下降。为了解决这一问题，我们引入了attention（注意力）机制。

假设有这样一则法律问题：“盗窃自行车，判刑多久？”，我们要从中抽取出关键词“盗窃罪”和“量刑标准”，attention机制在不同时间输入不同，如图3.9：

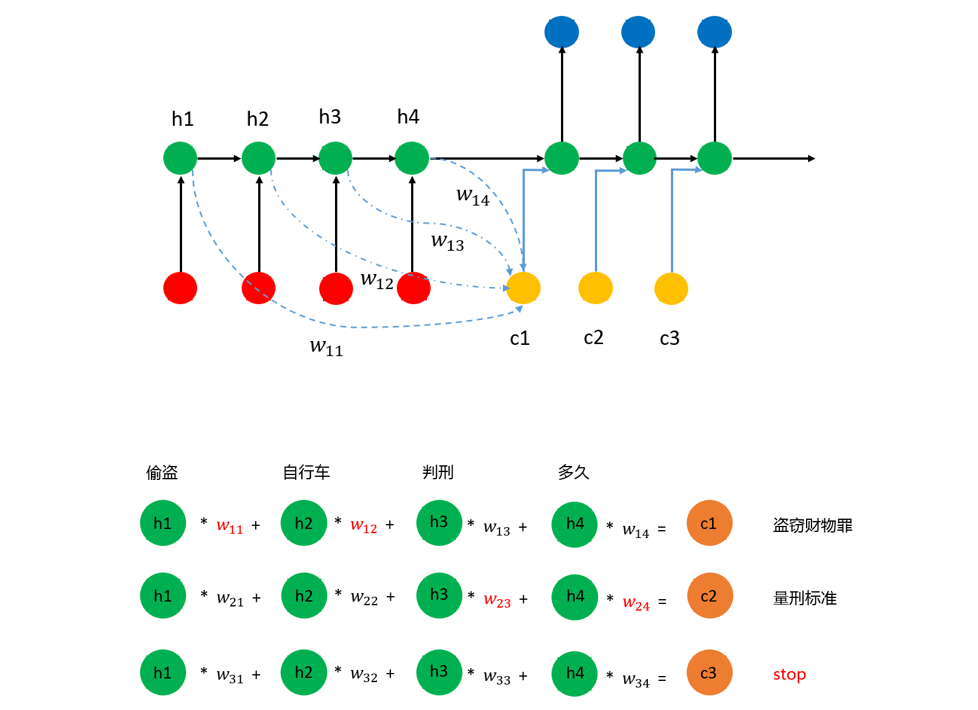


图3.9 attention机制

其中为输入序列对应的隐状态，~为Decoder在不同时间的上下文向量，为每个对应隐状态的权重。假设在对应“偷盗”、“自行车”、“判刑”、“多久”的编码，Decoder在时间时刻通过4个权重计算第一个上下文向量：

 (3-1)

再将与原始隐含层最后一个隐含向量连接构成一个新向量输入新的隐含层：

 (3-2)

当时刻以此类推得到，与共同构成一个新的隐含层，由于权重各不相同，就形成了Decoder对Eecoder不同输入的注意力，且权重越大表示Decoder在时刻越重视该输入。

经过上述构建模型，就可以将关键词提取任务转换为权重矩阵计算任务，假设表示Encoder状态，表示Decoder状态，表示注意力层输出的Decoder状态，那么时刻对Encoder隐含层每一个的权重的计算如下：

 (3-3)

一般使用一下方式计算：

  (2-4)

其中tanh函数在pytorch中通过调用，参数需要模型通过学习得到。

4 系统分析

4.1 数据需求分析

本系统需要从问法网（http://www.51wf.com）上收集大量法律问题语料，所收集的语料中每条数据必须包含：用户提出的法律问题，网站对该问题添加的标签。问法网的“问法知道”模块提供了数据支持，该网站的“问法知道”页面目前提供了从2008/12/30至2019/5/17期间所有用户提出的问题数据，其中分为“已解决法律咨询”和“待解决法律咨询”两类数据。

“已解决法律咨询”分类下的问题表示该问题已经得到专业律师的解答，用户根据其解答已经解决了所提出的法律问题，并且问法网也对该问题定义了标签。

“待解决法律咨询”分类下的问题表示该问题暂未得到专业律师解答，用户提出的问题暂未得到解决，问法网也暂时未对其添加标签。

经过分析，只有“已解决法律咨询”分类下的数据对本系统才是有意义的，故在收集“问法知道”数据时，应该过滤掉未带标签的法律问题。通过以上数据需求分析，本次共收集16万4千余条数据，能够涵盖4722个法律专业词汇，基本满足本系统对训练数据集的要求。

4.2 功能需求分析

本系统通过使用Pytorch框架建立一个基于序列到序列模型，在其中加入强化学习，从而生成一个模型来预测法律问题这类短文本中包含的关键词。具体功能模块包括：

（1）语料收集与预处理

（2）中文分词

（3）词向量生成

（4）搭建基于强化学习的seq2seq模型

（5）训练模型

（6）系统运行与测试

4.3 性能需求分析

由于在.机器学习.过程中涉及.到大量简单.的计算任务，使用CPU计算速率低.且容易造成.计算机死机导致.运行的程序奔溃。为了分析本系统实验.硬件环境中使用CPU与.GPU的差别，利用一套网上开源的.序列到序列模型项目，分别在本机CPU和GPU上运行。结果显示在4万条数据训练过程中，CPU耗时6小时，而GPU仅耗时11分钟。由实验结果.分析得出，本环境所使用的GPU计算速率为CPU的32倍，且训练耗时较短，基本满足对性能的要求。

5 系统实现

5.1 语料收集和预处理

5.1.1 数据爬虫

（一）链接获取

本实验数据从问法网（www.51wf.com）的“问法知道”页面抓取，首先分析该页面的链接格式，以当前最新问题为例，如图5.1所示：

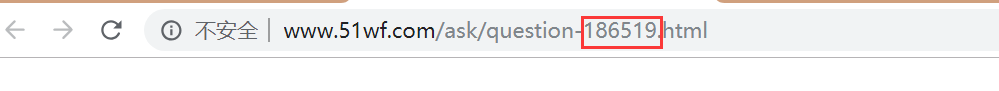


图5.1 链接格式

每个问题的链接是由一个类似于-186519的id标识的静态页面，这个id为网站后台数据库中该问题的唯一标识号，所以我们可以通过id[00000～186519]获取所有问题的链接。

（二）页面解析

有了所有链接之后，接下来的工作就是解析每个页面的数据。以图5.2为例：



图5.2 页面解析

由分析可知，该页面的结构简单易读，问题和每个标签都有明显标识。故可以BeautifulSoup库和lxml快速解析，从页面中提取出（问题-标签）数据。

5.1.2 数据预处理

在抓取数据并写入本地文件的同时，还需要对数据进行一定的预处理以便能够提供规范的语聊供模型训练，共有以下几步：

1. 去除标点符号

使用re库的sub()函数能够去除数据中的所有标点符号。

1. 标识问题和标签分隔符

在获取每条数据时，通过在“问题”和“标签”之间添加‘\t’字符加以分割。

1. 脱敏处理

数据中往往包含大量的阿拉伯数字，有可能对训练精度造成一定影响，故需要对数字进行一定的脱敏处理，在notepad++编辑器中利用正则表达式将所有数字替换为<num>。

5.2 中文分词

中文分词是本系统的首要任务，准确的分词结果能够使模型训练结果更加精确。本系统以jieba库作为语料中文分词工具，主要分为以下两步：

1. 添加法律词条扩展词典

在遇到某个比较长的法律词汇，在口语化时人们通常会使用缩写，比如“律师事务所”经常被缩写为“律所”。假设现在有一则法律问题中包含“违规律所”一词（即违规律师事务所），分词器由于字典限制有可能会将其分割为“违/规律/所”，这并不是句子真正的语义。为了解决这一问题，jieba支持添加自定义词典，以扩展词典里的词汇。

问法网目前提供了共213944个专业的法律词条数据，抓取这些词条并写入法律扩展词典中。图5.3展示了添加法律词条扩展词典前后的分词结果：

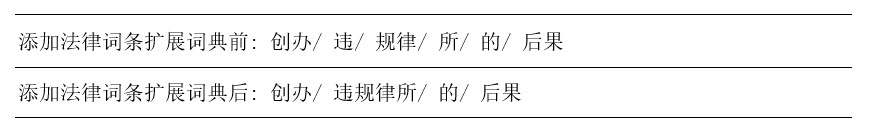


图5.3添加法律词条扩展词典前后的分词结果

（二）精确模式分词

使用jieba的精确分词模式可以将法律问题语句精准分词，分词结果样例见图5.4。

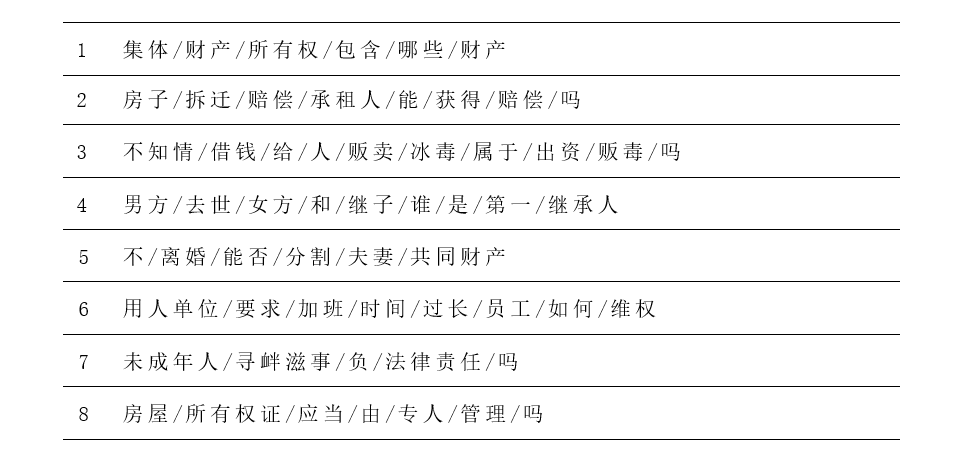


图5.4 分词结果样例

5.3 词嵌入

语料数据分词结束后，需要将所有词汇转换为词向量。gensim库中对word2vec的CBOW模型做了很好的封装，使用其封装的Word2Vec()方法训练模型，设置词向量维度为256。本次实验生成的.vector文件中包含的词向量共16722个，且每个词向量包含了其对应的语义信息，使用gensim库的most\_similar()方法加以测试，测试结果表明通过词向量能够找到与某个词关联最大的词或者近义词。使用Matplotlib展示部分词向量结果如图5.5所示。

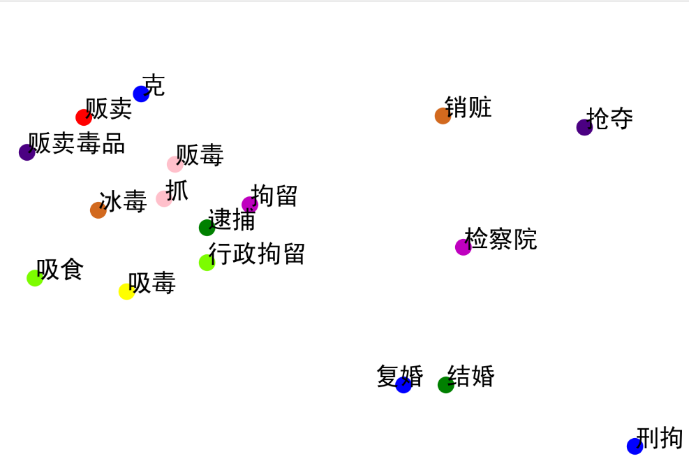


图5.5 部分词向量结果

5.4 基于强化学习的seq2seq模型

搭建基于强化学习的seq2seq模型是本系统的核心模块，本系统使用IBM在2018年开源的pytorch-seq2seq框架，该框架具有模块化和可扩展组件，用于seq2seq模型的训练、预测和检查点等，且该模型封装好attention机制，比较适合本系统需求。

5.4.1 Encoder-Decoder

（一）Encoder（编码器）

使用EncoderRNN()构建一个编码器RNN，将其应用于输入序列。其思路是定义词典大小、最大序列长度、隐含层特征个数等参数，同时传入之前已经训练好的词向量。该编码器的输入是语料库的原始序列的列表，列表长度为自定义的batch\_size（批量大小）。输出包括原始输入序列的特征张量，和包含隐状态h特征的张量。

1. Decoder（解码器）

使用DecoderRNN()构建一个解码器RNN，将其应用在seq2seq模型解码功能上。思路是将Encoder编码输出的结果作为输入，定义序列开始和结束标志符(<SOS>,<EOS>)作为解码的标志位进行解码。其输入包括Encoder中原始输入序列的特征张量和隐状态h特征张量，以及用于从RNN隐藏状态生成标记的激活函数（一般使用softmax）。

5.4.2 seq2seq模型

在定义好Encoder-Decoder后，将其共同构建为一个标准化的seq2seq模型。在编码过程中，编码器每一步输入的词构成输入序列 ,RNN将其转换为上下文向量*c*，编码过程如公式5-1：

 （5-1）

其中，为在第*t*步输入时的隐含状态，将编码后的结果输入解码器进行解码过程，解码器将上下文向量*c*展开为序列，解码过程如公式5-2：

 (5-2)

由式5-2可以看出，解码器在第*t*步时的隐含状态与、、*c*有关，输出与、、*c*有关，*f* ()与*g* ()都是激活函数。

5.4.3 attention机制

在Decoder的输出功能上应用attention（注意力）机制，输入为包含解码器输出特征的张量作为上一次的输出状态（output），以及包含编码器输入序列特征的张量作为上下文语义（context）。输出为包含解码器已注意到的输出特征张量（output\*），以及注意力权重张量（attn），计算公式如式5-3。

 (5-3)

通过定义一个nn.Module的子类并重写其*forward* ()方法实现式5-3算法，如图5.6。

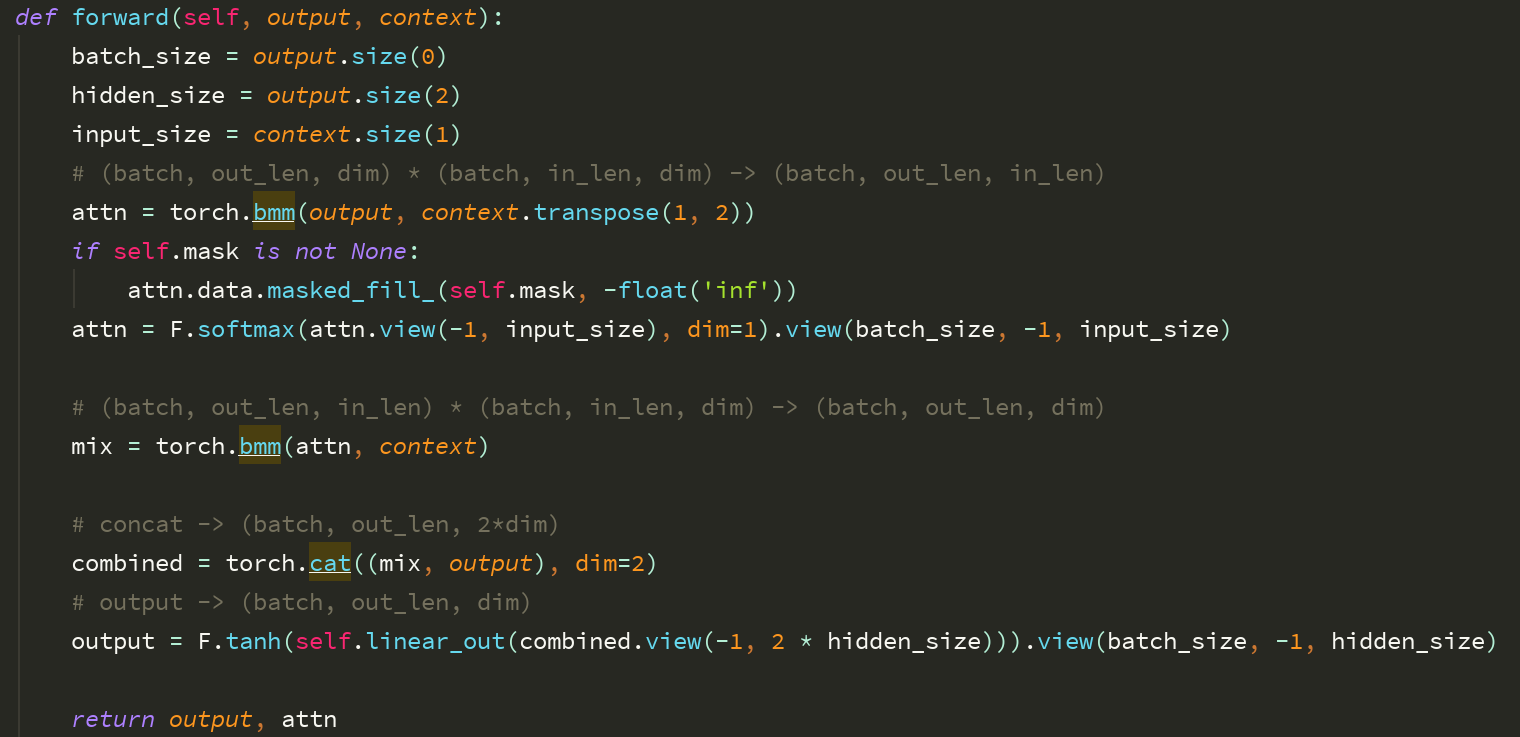


图5.6 实现attention算法部分代码

5.4.3 强化学习

强化学习（Reinforcement Learning）是指在机器学习的过程中，根据设定的奖惩规则不断尝试并从中找到规律的过程。当机器做出错误的动作时，给予一定的惩罚使之获得错误经验；当机器做出正确的动作时，给予一定的奖励使之获得激励。

强化学习主要包括四个因素：Agent（智体）、Environment（环境状态）、action（动作）、reward（奖励）。本系统强化学习的内容就是Agent根据当前的环境状态做出action，得到奖励后，将本身置为下一状态的过程，整个过程会不断循环直至终止状态。根据Markov（马尔科夫）决策过程，下一状态是由即将做出的action和当前状态共同决定的。本系统的Agent就是seq2seq模型，seq2seq模型生成的关键词作为action，生成的关键词用*k*表示。本课题属于文本处理问题，故选取基于策略梯度的PolicyGradient算法，该算法思路如图5.7。

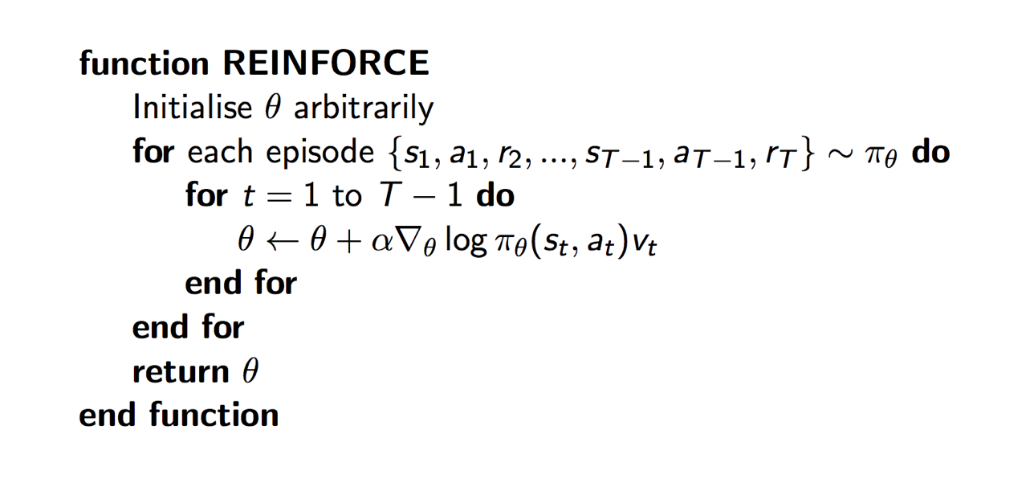


图5.7 PolicyGradient算法

设S为所有环境状态的集合，表示在*t-1*时刻的环境状态；设A为Agent执行所有可能action的集合，表示在*t-1*时刻Agent即将做出的动作；设R为决策过程中每一步奖励值的集合，表示在*t*时刻根据*t-1*时刻的*s*和*a*做出的奖励（reward），

故视为一个学习步骤。本系统为每一步学习步骤设置的奖励策略如式5-4：

 （5-4）

其中*K*表示目标关键词集合，表示动作*a*获得的奖励，为输出序列的长度。当Agent生成的关键词不在K中时（即生成了错误的关键词）或者输出了重复的关键词，我们给予0的奖励作为惩罚，这样reward（奖励）机制能够增加下一次好的action出现的概率，减少下一次不好的action出现的概率，达到强化学习的效果。

5.5 训练模型

6 系统测试

7 总结

。

参考文献

致谢

附录:程序主要源代码