

文章编号: 1003-0077 (2017) 00-0000-00

基于序列到序列模型的法律问题关键词抽取

曾道建^{1,2}, 童国维^{1,2}, 戴愿^{1,2}, 李峰^{1,2}, 韩冰³, 谢松县³

(1. 长沙理工大学 计算机与通信工程学院, 湖南 长沙 410114;
2. 长沙理工大学 综合交通运输大数据智能处理湖南省重点实验室, 湖南 长沙 410114;
3. 湖南数定智能科技有限公司, 湖南 长沙 410013)

摘要: 传统的关键词抽取算法不能够抽取那些没有在文本当中出现过的关键词, 因此传统的关键词抽取算法在抽取法律问题(短文本)的关键词任务上效果不佳。本文提出了一种基于强化学习的序列到序列(seq2seq)模型从法律问题中抽取关键词。首先, 编码器将给定法律问题文本的语义信息压入一个密集矢量; 然后, 解码器自动生成关键词。因为在关键词抽取任务中, 生成的关键词的前后顺序无关紧要, 所以引入强化学习来训练我们的模型。所提出的模型结合了强化学习在决策上的优势和序列到序列模型在长期记忆方面的优势, 在真实数据集上的实验结果表明, 带有强化学习的序列到序列模型在关键词抽取任务上有较好的实验效果。

关键词: 关键词抽取; 序列到序列模型; 强化学习

中图分类号: TP391

文献标识码: A

Keyphrase extraction for legal questions based on sequence to sequence model

Daojian Zeng^{1,2}, Guowei Tong^{1,2}, Yuan Dai^{1,2}, Feng Li^{1,2}, Bing Han³, Songxian Xie³

(1. School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, Hunan 410114, China;
2. Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha, Hunan 410114, China;
3. Hunan Date-driven AI Technology Co. Ltd., Changsha, Hunan 410114, China)

Abstract: Traditional keyphrase extraction algorithms cannot extract keyphrases that have not appeared in the text. Therefore, traditional algorithms perform poorly on the task of extracting keyphrases from legal questions (short texts). In this paper, we propose a sequence-to-sequence (seq2seq) model based on reinforcement learning to extract keyphrases from legal questions. First, the encoder compresses the semantic information in the given question text into a dense vector. Then, the decoder automatically generates the keyphrase based on the encoder output. As the order of the generated keyphrases has no effect on the final effect, we use reinforcement learning to train our model. The proposed model combines the strengths of reinforcement learning in decision-making and the advantages of seq2seq models in long-term memory. The experimental results demonstrate that the Seq2Seq model with reinforcement learning can achieve the state-of-the-art performances on real-world data sets.

Key words: keyphrase extraction; sequence-to-sequence; reinforcement learning

收稿日期: ; 定稿日期:

基金项目: 国家自然科学基金(61602059); 湖南省自然科学基金(2017JJ3334);
湖南省教育厅科学研究项目(16C0045); 模式识别国家重点实验室开放课题基金(20170007)

0 引言

关键词通常是一段简短的和总结性的内容，它能够描述较长文本中的主题信息^[1]。高质量的关键词能够为用户提供高度浓缩和有价值的信息。关键词抽取是自然语言处理中一个重要的任务，它在信息检索，问答系统，文本摘要和文本分类等任务上发挥着重要作用。由于关键词抽取具有重大现实意义，许多研究者已经使用不同的方法对自动关键词抽取进行了研究^{[2][3][4]}。

由于科学出版物容易被公众获取，所以许多科学出版物数据集经常被用作关键词抽取算法的测试^[5]。这些研究通常集中在从文档和文章中抽取关键词，然而这些文档的特点是一般有较长的篇幅。大多数已经存在的关键词抽取算法通常使用两个步骤^{[3][6]}来解决这个问题：第一步是将要抽取关键词的内容分为多个文本块，这些文本块作为候选关键词；第二步是对候选关键词按照对文本内容的重要性来进行排序。

随着网络信息的快速发展，在微博和推特上，短文本推文逐渐成为人们的主要信息来源。如何从海量的网络短推文中快速获得所需要的关键信息变得越来越重要，一些研究人员^{[7][8]}已经开始研究如何从短文本（例如推特）中抽取关键词。

本文的研究是解决从法律问题中自动抽取关键词的问题，法律问题属于短文的类别。图 1 是一个从法律问题中抽取关键词的例子。法律领域问答系统能使人们更容易的获得法律信息。关键词有助于问答系统更好更快的理解问题背后的用户意图，所以关键词抽取技术对法律问题解答系统十分重要。

相比于从长文本中抽取关键词，从短文本中抽取关键词的研究更加困难。首先，在短文本中，许多语言学特征和统计特征无法使用。传统的关键词抽取方法是基于词出现和共现的次数来判断词的重要性，这类算法无法获取内容中的隐含语义。其次，两步关键词抽取方法（候选关键词生成和候选关键词排序）只能抽取那些已经在原文本中出现过的短语作为关键词。然而在法律问题的短文本中对应的关键词并不会在原文本中一模一样的出现，例如在图 1 的例子中，关键词受益人、信用卡诈骗、抢劫罪购构成标准和抢劫罪量刑标准并没有在源问题文本中出现。

综上所述，当关键词在文档中以轻微不同的连续顺序或者同义词形式出现时，这类传统的方法将不能准确的抽取关键词。然而在法律问答系

统中，问题是由普通用户提交的而不是由法律专家提交的，由于专业领域的限制，普通用户提交的问题文本口语化程度比较高，他们不会使用如图 1 所示的受益人、信用卡诈骗、抢劫罪构成标准和抢劫罪量刑标准等专业的法律词汇。用传统的基于两步的算法直接从原文本中抽取关键词将不能生成正式的法律术语。针对前文提到的这些问题，本文提出了一种新的方法根据问题的语义信息来抽取关键词的新方法。

问题1	被保险人死亡，保险金是否属于遗产范围？
关键词1	保险金、遗产范围、受益人
问题2	朋友冒用了他人信用卡，触犯了何种法律，量刑如何？
关键词2	冒用他人信用卡、信用卡诈骗
问题3	多次抢劫的回构成抢劫罪吗？会判多久呢？
关键词3	抢劫罪构成标准、抢劫罪量刑标准

图 1: 关键词抽取的例子

该方法将关键词抽取看作生成问题而非简单的抽取问题。对于一个法律问题，我们人类首先阅读问题文本，获得对问题文本内容的基本理解，然后将问题文本内容总结为一个个的关键词。这些关键词可能是没有在原文本中出现过的词。除了语义理解之外，我们人类也会根据句法特征来选取文本中最重要的部分作为关键词。因此，我们需要一个模型来获取语义信息，然后自动生成关键词。

受到 Cho 等在机器翻译^[9]上和 Vinyals 在句法分析^[10]上运用基于神经网络的序列到序列模型取得很好的实验效果的启发，我们提出了一种基于强化学习的序列到序列模型来对法律问题自动生成关键词。在我们所提出的模型中，用户提交的法律问题首先被编码器转换成固定长度的向量。然后，解码器解码该向量获得关键词。通常的序列到序列模型使用交叉熵进行训练时，是按照顺序来生成关键词，但是在关键词抽取任务中，关键词的前后顺序对于最终结果不会有太大影响，因此我们用强化学习^[11]来训练我们的模型。

1 相关工作

关键词以简洁概括的短语描述文档，目前已经有许多关键词抽取算法。这些算法通常分为两个步骤，第一步是利用一些启发式规则生成关键词候选集。因为生成的候选词要在之后进行过滤，

为保证最终被抽取的关键词的准确性, 所以在这个步骤中会生成大量的候选关键词。生成关键词候选集的主要方式有去除停用词, 抽取指定词性的词^[3]错误!未找到引用源。 (例如名词, 形容词等), 抽取在 Wikipedia 等重要语料库中的 n-gram 和按照事先制定好的规则抽取 n-gram 或者名词短语^[4]等方法。

第二步是计算在候选关键词集中的每个候选关键词在文本中作为准确关键词的可能性。选取排名最高的候选关键词作为准确关键词。在此步骤中广泛使用的方法是有监督和无监督机器学习方法。在有监督的机器学习方法中, 抽取关键词任务被转化为二元分类问题。有监督方法就需要有人工标注好的数据集, 利用已经标注好的训练集训练分类器, 同时利用训练好的分类器对训练集中的关键词进行抽取。抽取的关键词是训练集中的标注好的准确关键词就作为正例, 如果不是就作为负例, 然后正例和负例共同训练得到一个适用于关键词抽取的最终分类器。许多训练方法都已经用在训练分类器上, 例如 Frank 等采用朴素贝叶斯训练分类器^[2]和 Turney 等采用决策树训练分类器^[1]。有监督关键词抽取方法还需要特征训练分类器, 这些特征主要分为两大类: 文本内部特征和文本外部特征, 文本内部特征包括统计特征, 结构特征和句法特征。文本外部特征包括 Wikipedia 等语料库中的词频, 搜索引擎中查询的次数和词之间的语义关联等。无监督的方法有 Mihalcea 和 Tarau^[12]提出的计算候选关键词之间的关联性基于图的排序方法和 liu 错误!未找到引用源。等提出的利用聚类的 KeyCluster 方法。

这些采用机器学习的算法例如 TF-IDF 和 TextRank^[12]都使用了大量的文本内语言学和统计学的特征。然而我们在抽取法律问题(短文)关键词的任务上, 文本中只含有极为少量的这类特征。

有一些学者已经开始研究如何从短文中抽取关键词, 例如 Zhang^[8]等提出了一种联合循环神经网络模型对短文本进行关键词抽取。但是他们所提出的模型并不能抽取原文本中没有出现过的关键词。在短文本中, 并不是所有的关键词都会在原文中出现。

为了解决这个问题, 我们把关键词抽取转化为关键词生成。我们的方法基于序列到序列模型, 同时在解码器网络中加入强化学习来优化训练。我们的模型能以序列的形式生成关键词, 所以它能够总结那些没有在原文本中出现过的词语来作为关键词。

2 方法

2.1 方法概述

典型的基于循环神经网络(Recurrent Neural Network, RNN)的编码器-解码器通常由两个 RNN 组成: 一个作为编码器, 另一个作为解码器。在我们的模型中, 对编码器和解码器进行联合训练以最大化给定源序列的目标序列的奖励如图 2 所示, 我们提出的方法是在解码器中用强化学习插入编码器-解码器框架。整个系统的输入是一个句子, 它首先被编码器转换成相应的向量表示。然后, 这些向量表示被馈送到解码器以生成关键词。正如介绍中所提到的, 关键词的顺序对于关键词抽取这一任务是无关紧要的, 我们只需关注生成关键词的正确性。因此, 我们使用强化学习来优化外部奖励模型而不是提供每个时间步骤的监督标签。

2.2 基于循环神经网络的编码器-解码器

循环神经网络(Recurrent Neural Network, RNN)是一种反馈型的神经网络。RNN 的输出不仅与网络的权重和当前时刻的输入有关, 还和之前的输入有关。在 RNN 的结构中, 隐含层中的神经元互相连接, 它的输入不但有当前时刻的输入还有上一个时刻隐含层神经元的输出。在训练 RNN 的过程中, 会有长时间的传播, 导致最后的求导结果趋于零, 从而出现梯度消失的问题。所以在实践中, RNN 还有一些门变体的形式例如 LSTM (Long Short-Term Memory)^[13]和 GRU (Gated Recurrent Unit)^[14]。LSTM 隐含神经元通过不同的门来控制记住信息和忘记信息, 所以 LSTM 能够克服 RNN 的缺点, 从而学习到长时间的信息。RNN 的另一种变体是双向循环神经网络(Bi-directional Recurrent Neural Network), 它能够连接正时间方向和负时间方向两种隐含状态。

基于 RNN 的编码器-解码器是主要用于序列到序列学习^[9]。在编码器-解码器的框架中, 分为编码过程和解码过程。在编码过程中, 每个时刻向编码器(Encoder)输入一个词, 隐含层就会根据公式(1)而变化, 当输入到最后一个词 x_s 时, RNN 编码就会最终将源序列 $X = [x_1, \dots, x_s]$ 转换成编码表示 c 。因为 RNN 会把前面每一步的输入信息都保存, 所以向量 c 能够包含源序列 X 的所有信息。我们可以用公式(1)来描述编码过程:

$$h_t = f(x_t, h_{t-1}); c = \phi(h_1, \dots, h_t) \quad (1)$$

其中 h_t 是时间 t 时 RNN 隐藏状态, c 可以被看作 X 抽象表示的上下文向量。在源序列 X 被编码后, 编码过程结束。进入到解码过程, RNN 解码器就

将被压缩上下文向量 c 展开到目标序列 $Y = [y_1, \dots, y_s]$ 中, 解码器 (Decoder) 在 t 时刻的隐含状态 s_t 是由 s_{t-1}, y_{t-1}, c 共同决定的, y_t 是

s_t, y_{t-1}, c 由共同决定。解码过程可以通过公式 (2) 来描述:

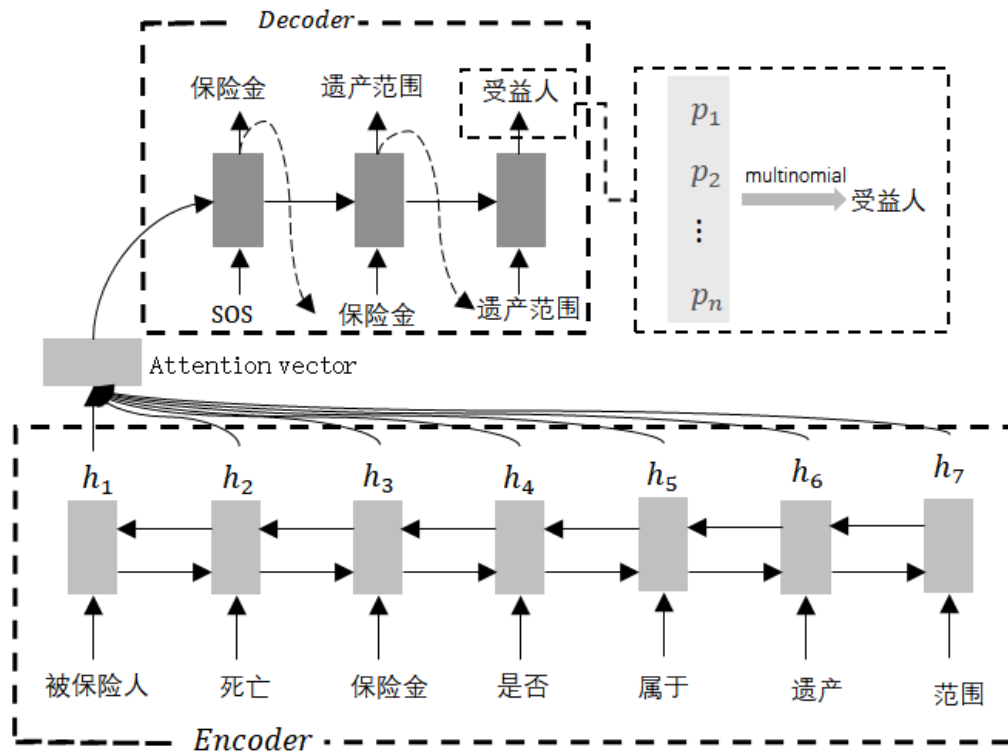


图 2: 基于序列到序列的关键词抽取的框图

$$S_t = f(y_{t-1}, s_{t-1}, c)$$

$$p(y_t | y_{<t}, X) = g(s_t, y_{t-1}, c) \quad (2)$$

其中 S_t 是时间 t 时 RNN 的隐含状态, y_t 是时间 t 处的预测目标词, f, g 都是激活函数, 其中 g 函数一般是 softmax。

在解码器的设计上, 每个时刻都是用了相同的上下文向量 c , 有研究人员提出可以在不同时刻输入不同的上下文向量。引入了注意力机制的编解码器, 就出现了引入了注意力机制的编解码器, 也就是将向量 c 改为 $c_{t'}$, 表示 t' 时刻的上下文向量。那么在 t' 时刻, 解码器的隐含状态为:

$$s_{t'} = f(y_{t'-1}, s_{t'-1}, c_{t'}) \quad (3)$$

取编码器隐含状态 h_t 的加权平均来设计不同的上下文向量 $c_{t'}$, 即:

$$c_{t'} = \sum_{t=1}^T a_{tt'} h_t \quad (4)$$

其中是 $a_{tt'}$ 表示权重, 与当前时刻编码器的隐含状态 h_t 以及上一个时刻的解码器隐含状态 $s_{t'-1}$ 有关, 即:

$$a_{tt'} = \text{softmax}(a(s_{t'-1}, h_t)) \quad (5)$$

选取不同的函数 a , 可以得到不同的注意力机制。

2.3 强化学习

强化学习是环境状态映射到动作的学习, 目的是使 agent 在与环境交互过程中获得最大的累计奖励。强化学习的过程就是 agent 根据当前所

处的状态, 做出动作, 得到回报, 转移到新的状态。这个过程会一直重复持续下去, 直到到达终止状态。这个过程是马尔科夫决策过程, 即下一个时刻的状态有且仅和当前时刻所处的状态和将要做出的动作有关。在本文的任务中, Seq2Seq 模型被认为是一个 agent。我们用 p 来表示从 agent 生成的关键词。一个关键词抽取系统可以表示为由 agent 生成的一系列关键词。我们将生成的关键词视为根据策略执行的动作, 这个策略是由编解码器的循环神经网络模型定义的。优化网络参数以最大限度地提高策略搜索的累计奖励。

强化学习算法主要分为基于值函数的算法^[15]和基于策略梯度的算法^[16]。基于策略梯度的算法比基于值函数的 Q-学习算法更适合我们场景。因为在改变目标并调整为最大化总奖励的策略之前, 可以使用已经产生合理响应的 MLE 参数初始化基于 RNN 的编码器 - 解码器。另一方面, Q-学习算法是直接估计每个行动的预期奖励, 这可能与 MLE 目标相差数量级, 因此使得 MLE 参数不适合初始化。基于策略梯度的算法能够直接优化策略的预期总奖励, 并直接在策略空间中搜索最优策略。

本文中的序列决策问题的组成部分 (状态,

动作, 奖励等) 在以下小节中进行了总结。

2.3.1 动作

A 为 agent 可以执行的动作的集合, 动作 $a_t \in A$ 表示在 t 时刻 agent 做出的动作。动作 a 是问题的关键词序列。由于可以生成任意长度的序列, 动作空间是无限的。

2.3.2 状态

P 为所有状态的集合, $p_{i-1} \in P$ 表示在 t 时刻 agent 所处的状态。当前的状态由先前生成的关键词 p_{i-1} 决定。通过将 p_{i-1} 送到 RNN 编码器模型中, 之前的解码被进一步转换为矢量表示^[17]。

2.3.3 策略

策略是在当前状态 p_t 选择做出动作 a_t , 由于执行了动作 a_t , agent 会转移到下一个状态 p_{t+1} , 同时会得到来自环境的奖赏 $r(a)$ 。策略采用 RNN 编码器-解码器 (即: $P_{RL}(p_i|p_{i-1})$) 的形式, 并由其参数定义。

2.3.4 奖励

与监督学习计算每一步骤的损失不同, 我们设计评估标准来计算每个行动获得的奖励。最终奖励函数描述如下:

$$r(a) = \frac{b_r}{N_s} \quad (6)$$

其中 $r(a)$ 表示动作 a 的奖励, N_s 是输出序列的长度, b_r 的定义如下:

$$b_r = \begin{cases} 1 & \text{if } a \in T \text{ and not duplicate} \\ 0 & \text{else} \end{cases} \quad (7)$$

其中 T 表示目标序列集合。

即使产生的关键词序列顺序与训练集的顺序不同, 奖励函数也会为产生正确关键句的动作给出高分。由于输出序列的长度是不确定的, 为了使奖励标准化, 我们将最终奖励除以序列的长度。如果模型生成重复的关键词, 就会对动作添加惩罚。

3 实验

本节介绍我们的模型在一个真实存在的数据集上进行关键词抽取的实验, 并对比和分析实验结果。我们的模型使用 PyTorch 工具在 NVIDIA TITAN X GPU 上训练, 使用 Adam^[18] 学习规则更新实验配置中的梯度。

3.1 数据集和评估指标

3.1.1 数据集

实验所用的数据集 WF 是从一个主流的中国法律社区问题答案网站¹获得的, 该网站包含法律问题和由用户协作注释的关键词。我们在表 1 中

给出了该数据集的详细统计信息。

表 1: 数据集的统计信息。

数据集	D	W	T	$\overline{N_d}$	$\overline{N_a}$
WF	60812	18000	10302	11.9	3.2

D 、 W 、 T 、 $\overline{N_d}$ 和 $\overline{N_a}$ 分别是例子的数量, 问题的词汇, 关键短语的词汇, 每个问题的平均单词数量以及每个例子的关键短语的平均数量。

3.1.2 评估指标

我们使用准确率来度量模型的精确度, 用召回率来度量模型的完整性和 F 值 (精度和召回率的调和平均数), 以评估我们的关键词抽取方法的性能。对于资源, 用 T_o 表示准确的关键词, T_e 表示抽取的关键词, 用 $T_o \cap T_e$ 表示正确抽取的关键词。精确度, 召回率和 F 值定义如下:

$$p = \frac{|T_o \cap T_e|}{|T_e|}, r = \frac{|T_o \cap T_e|}{|T_o|}, F = \frac{2pr}{p+r} \quad (8)$$

3.1 设置

在本文实验中, 使用 GRU 作为模型单元。单元数量设置为 256; 批量大小为 256。最大时间步长 T 为 10, 所以每个问题最多可预测 10 个关键词。当在验证集中得到最佳结果时, 停止训练。

3.3 方法比较

我们实现几种基线方法来和我们提出的方法进行对比, 以评估本文所提出的方法的有效性。TF-IDF 和 TextRank^[12]: 这两种方法从问题中抽取可能的关键短语并对它们进行排名, 随着问题中的关键词短语频率增加而增加分数, 并随着语料库中的关键词短语频率而降低。

RNN 和 CopyRNN: 这两种方法基于序列到序列模型来抽取可能的关键词。RNN 是指具有交叉熵损失的基本序列到序列模型。CopyRNN^[19] 在 RNN 中引入了一个拷贝机制

表 2: 关键词抽取算法的实验结果

	p	r	F
TF-IDF	0.4665	0.5195	0.4916
TextRank	0.5977	0.6231	0.6101
RNN	0.7609	0.7885	0.7745
CopyRNN	0.7551	0.7618	0.7584
Ours	0.7784	0.8066	0.7923

3.4 实验结果

表 2 显示了不同的关键词算法的实验结果。根据这些结果, 可以做出以下结论:

基于序列到序列的方法比传统的两个步骤方法更好的性能。这种改进可以通过以下事实来解释: 准确标准中的许多关键词不匹配原文本的任何连续子序列。这背后的另一个原因可能是中文

¹ www.51wf.com

分词存在一些错误，而关键词可能由多个中文词组成。

值得强调的是 RNN 比 CopyRNN 更好。这意味着引入了拷贝机制对模型的性能的提高没有帮助。通过对表中的数据分析，发现该数据集中的许多关键短语不能通过复制模式生成。

在精确度，召回率和 F 值方面，我们的模型表现明显优于其他所有被评估的方法。在与所有比较方法上，我们提出的方法都达到了最佳性能。值得注意的是，我们使用强化学习而不是交叉熵学习。结果表明所提出的方法是能够有效用于关

键词抽取。

案例研究与错误分析：表 3 给出了我们的模型生成的关键词的一些例子以及测试集中问题的关键词。在问题 ID1 里，关键词是信用卡诈骗、信用卡而生成的关键词是信用卡、信用卡诈骗。在问题 ID2 里，关键词是交通肇事罪、危险驾驶罪而生成的关键词是危险驾驶罪、交通肇事罪。这表明所提出的模型能够忽略关键词的顺序。分析抽样的例子，主要的错误原因如下：1. 对问题理解的不充分（ID3）；2. 生成的关键词不完整（ID4）；3. 标注的关键词不正确（ID5）。

表 3: 关键词生成的例子

ID	问题	关键词	生成的关键词
1	冒用他人信用卡法律如何量刑?	信用卡诈骗 信用卡	信用卡 信用卡诈骗
2	危险驾驶罪和交通肇事罪什么联系和区别?	交通肇事罪 危险驾驶罪	危险驾驶罪 交通肇事罪
3	购买的宝马车是套牌车报警会以诈骗罪立案吗?	诈骗罪立案	合同诈骗罪
4	敲诈勒索罪的概念及构成是什么?	敲诈勒索罪构成 敲诈勒索罪	敲诈勒索罪
5	涉嫌挪用公款罪被关北京市第一看守所此罪在刑法及司法解释上是怎样规定的?	北京市第一看守所	北京市第一看守所 挪用公款罪

4 结论

在本文中，我们提出了一个基于神经网络的序列到序列模型来抽取法律问题中的关键词，并在序列到序列模型中引入了增强学习，使模

型能够在序列级别上预测短语。在真实法律领域的数据集上，实验对比分析了主流方法，证明我们提出的方法是一种有效的关键词抽取技术，并且通过引入强化学习，我们可以生成准确性很高的关键词。

参考文献

- [1] PD Turney. Learning Algorithms for Keyphrase Extraction[J]. Information Retrieval, 2002, 2(4): 303-336.
- [2] Frank E, Paynter GW, Witten IH, et al. Domain-Specific Keyphrase Extraction[C]// International Joint Conference on Artificial Intelligence, 1999: 283-284.
- [3] Liu Z, Li P, Zheng Y, et al. Clustering to find exemplar terms for keyphrase extraction[C]// Proceedings of the 2009 Conference on Empirical Methods in Nat-

- ural Language Processing: Volume 1-Volume 1. Association for Computational Linguistics, 2009: 257-266.
- [4] Medelyan O, Frank E, Witten IH. Human-competitive tagging using automatic keyphrase extraction[C]// Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3. Association for Computational Linguistics, 2009: 1318-1327.
- [5] Hasan KS, Ng V. Automatic Keyphrase Extraction: A Survey of the State of the Art[C]// Meeting of the Association for Computational Linguistics, 2011: 1262-1273.
- [6] Wang M, Zhao B, Huang Y. PTR: Phrase-Based Top-

ical Ranking for Automatic Keyphrase Extraction in Scientific Publications[C]// International Conference on Neural Information Processing. Springer, Cham, 2016: 120-128.

- [7] Bellaachia A, Al-Dhelaan M. Ne-rank: A novel graph-based keyphrase extraction in twitter[C]//Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on IEEE, 2012, 1: 372-379.
- [8] Zhang Q, Wang Y, Gong Y, et al. Keyphrase Extraction on Using Deep Recurrent Neural Networks on Twitter [C]//Conference on Empirical Methods in Natural Language Processing, 2016:836-845.
- [9] Cho K, Merriënboer B V, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J].Computer Science, 2014.
- [10] Vinyals O, Kaiser Ł, Koo T, et al. Grammar as a foreign language[C]//Advances in Neural Information Processing Systems, 2015: 2773-2781.
- [11] Ranzato M A, Chopra S, Auli M, et al. Sequence level training with recurrent neural networks[J]. arXiv preprint arXiv:1511.06732, 2015.
- [12] Mihalcea R, Tarau P. TextRank: Bringing Order into Texts[J].Emnlp, 2004:404-411.
- [13] Hochreiter S, Schmidhuber J. Long short-term memory[J].Neural Computation, 1997, 9(8):1735-1780.
- [14] Chung J, Gulcehre C, Cho K H, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. Eprint Arxiv, 2014.
- [15] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. arXiv preprint arXiv:1312.5602, 2013.
- [16] Sutton RS. Policy Gradient Methods for Reinforcement Learning with Function Approximation[J].Submitted to Advances in Neural Information Processing Systems, 1999, 12:1057-1063.
- [17] Li J, Galley M, Brockett C, et al. A diversity-promoting objective function for neural conversation models [J].arXiv preprint arXiv:1510.03055, 2015.
- [18] Kingma DP, Ba J. Adam: A method for stochastic optimization[J].arXiv preprint arXiv:1412.6980, 2014.
- [19] Meng R, Zhao S, Han S, et al. Deep keyphrase generation[J].arXiv preprint arXiv:1704.06879, 2017.



曾道建 (1985—), 博士, 讲师, 主要研究领域为自然语言处理。
E-mail: zengdj@csust.edu.cn



童国维 (1993—), 硕士研究生, 主要研究领域为自然语言处理。
E-mail: tongguoWei@stu.csust.edu.cn



戴愿 (1995—), 硕士研究生, 主要研究领域为自然语言处理。
E-mail: daiyu@stu.csust.edu.cn