互联网文本的法律法条引用跟踪分析

信息学院 苏展 2020000122

一. 任务介绍

我们致力于提出一种能够有效解决法律相关特有问题的高准确度的法律法条引用信息抽取算法。由于互联网文本和法律条文并不属于同一类型的文本,两者在篇幅、用词、表达规范方面具有显著的区别,互联网文本中通常只有一小部分包含法律内容,而传统的文本匹配方法中,根据词汇相似度判别的词袋模型(BOW)和未考虑词序的向量空间模型(VSM)无法简单地应用于引用法条的抽取问题。近来,深度匹配模型取得较多进展,然而这类模型训练需要大量标注的文本数据,而且由于一部法律往往有上百条法条,能够匹配成功的法条数绝大部分在 1 至 2 条,正例与负例的比例相差太大,直接标注全部法条数据得到的训练集数据不均衡,文本和法条的匹配过于稀疏,难以直接训练出有效匹配的模型。

基于前述任务特点,我们提出了一种融合确定有限自动机(DFA)、特征词序列和深度匹配的混合层次抽取模型 DS-LSTM。首先使用 DFA 从文本中进行初步匹配,通过法律名和法条号来初步筛选和过滤可能匹配的法条。对于已经抽取了法条号的文本,进一步通过比较各个历史版本的法条,做到精确分类;对于没有获取法条号的文本,通过构建特征词序列来表示文本,计算序列之间的相似度来衡量文本和法条的匹配程度,据此给出候选的法条,再使用 MV-LSTM 模型计算文本与候选法条之间的语义相似度,根据语义相似度来判断法条与文本是否匹配。

本研究的主要贡献是根据法律法规文本匹配问题,提出了融合多种抽取方法的混合层次抽取模型,而不是单纯使用深度模型或者 DFA 进行简单的匹配。虽然目前存在许多效果较好的深度匹配模型,但是在文本的法律法条抽取任务中,待匹配的文本引用的法律法条数量不同、表达方式不同造成需要大量各种类型训练数据的问题,以及待匹配的法律法条数量众多,实际匹配数量只有一两条法条造成的数据倾斜问题,直接使用深度文本匹配模型并不能实现高精度的抽取质量,而引入 DFA 进行抽取可以很好地缩小可能匹配的法律以及法条范围。同时根据识别结果区分出多种不同类型的文本,可以有效地避免训练

数据倾斜的问题。使用 DFA 抽取出文本中可能包含的多部法律名以及法条号，同时将文本按照法律以及法条出现的位置分割成文本片段，避免了多部法律在匹配过程中相互影响的问题。我们的方法在法律以及法条级别匹配结果的F_1分别达到 0.97 和 0.92。

混合层次抽取模型 DS-LSTM 的主要模块将在以下部分介绍。

二．核心算法介绍

2.1 整体模型概览

对文本的法律法条抽取匹配过程如图 1 所示，待匹配的文本需要经过文本预处理模块，去除文本中包含的特殊字符和停用词，之后用 2.2 中定义的 DFA 在处理后的文本上识别法律和法条，针对多版本法律以及隐式引用的问题，我们设计了两个处理的分支，如果在法条级别识别成功，说明文本是显示引用的法条，则进入历史版本区分模块；若法条级别识别失败，说明文本可能是隐式引用法条，因此就进入特征词序列匹配和深度匹配串联的层次抽取模块，进一步根据文本片段的内容判断可能引用的法条。其中特征词序列匹配模块利用 DFA 识别得到的法律名，在该法律的所有法条中计算序列的相似度，根据相似度筛选出可能匹配的法条集合，由深度文本匹配模型进一步匹配，最后得到匹配结果。

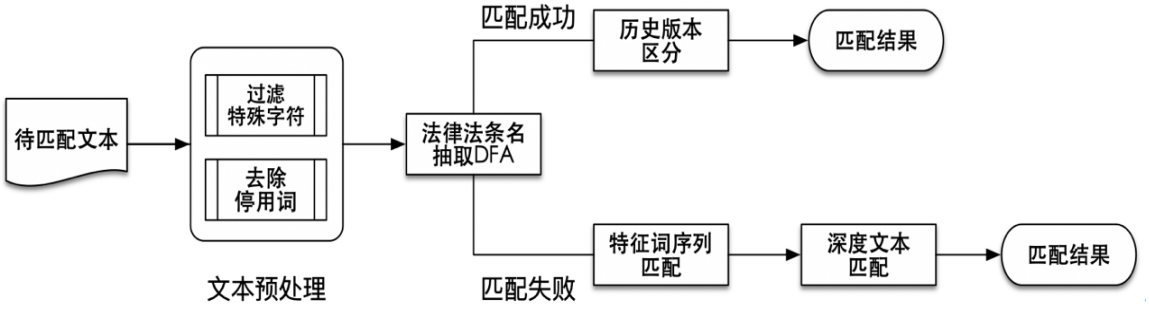

图 1 法律法条抽取流程

从整体模型来看，特征词序列匹配模块和深度匹配模块之间的关系是互补的以及不可代替的。之所以使用特征词序列匹配以及深度文本匹配两个模块，一方面是因为特征词序列匹配模块倾向于匹配引用法律原文的文本，对于表达较为跳跃以及对法律原文概括的文本，特征词序列匹配的方法无法很好地适用；另一方面由于一部法律通常有上百条法条，如果匹配上的法律还有多个历史版本，则需要在数百条法条中寻找匹配的法条，而实际上真正能够匹配上的法条只有一两条，

因此待匹配的法条数据中正例负例的比例差距太大，通常对不均衡数据的训练方法有对多数类的数据进行下采样、对少数类的数据进行上采样或者修改目标函数，本文使用特征词序列匹配的方法过滤掉绝大部分毫不相关的法条，提供少量比较相关的候选法条作为深度文本匹配模块的输入，缩小了正例与负例之间的比例差距。

2.2 法律法条识别确定有限自动机

定义 DFA M=$(S, \Sigma, \delta, S_0, F)$，法律名集合 A={‘宪法’,‘刑法’,…}，法条名集合 B={‘第(.*)条’,‘第(.*)条到第(.*)条’,‘第(.*)条至第(.*)条’}，有穷字母表 Σ=AUB ,状态集 S=$\{S_0, S_1, S_2\}$,终态集 F=$\{S_1, S_2\}$，δ为映射函数。

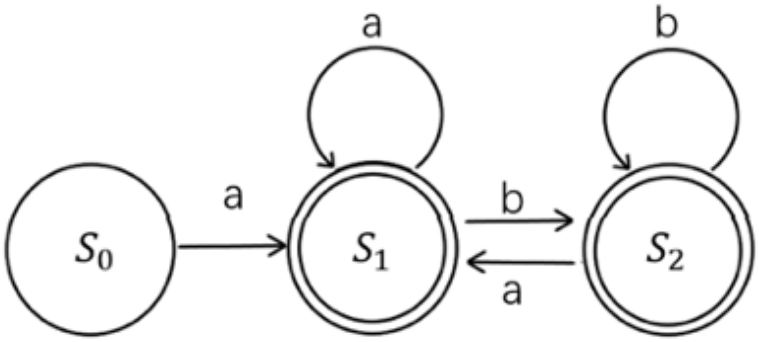

图 2 法律法条识别 DFA 的状态转换图

如图 2 所示，$a \in A, b \in B$，S_1, S_2分别表示识别末尾是法律名和法条名的终态，从$S_0 \rightarrow S_1$对应的是法律的识别过程，从$S_1 \rightarrow S_2$对应的是法条的识别过程。

在实际处理过程中发现，文本中包含的法律名通常不是法律的全称而是省略的叫法，例如用“《刑法》”代替“《中华人民共和国刑法》”，因此需要在法律名集合中加入法律的别名作为识别的关键词。此外，文本数据中包含法条的条数经常用阿拉伯数字和汉字混合表示，例如“第 2 条”和“第二条”;还有如“第 3 至 5 条”区间描述等情况不再赘述。

2.3 历史版本区分

历史版本区分步骤是在已知文本对应的法律名和法条号的基础上，判断对应法律的不同历史版本哪一个更符合文本片段的叙述。一部法律若存在多个历史版本，则它的某个法条在历史版本中一共有三种情

况：（1）与历史版本对比该法条相同（2）历史版本中该法条被修改（3）历史版本中没有对应的法条。

对于第一种和第三种情况没有区分的必要，而对于第二种情况，由于法条内容存在不同之处，所以尝试从不同的版本中选出最具有法条特点的词。我们在区分不同历史版本的法条时，通过计算 TF-IDF 选取出每个不同版本法条的特征词，根据特征词在待匹配文本中出现的频次，加权计算的每个法条的匹配度。选取法条特征词的时候以法条为单位，取出该法律所有历史版本中符合的法条进行分词、统计词频。将初步筛选后符合的法条当做文档 d,则满足条件的法条集合$D = \{d_1, d_2, \dots, d_m\}$,所有文档分词后形成的词集合$W = \{w_1, w_2, \dots, w_n\}$,则可计算第$i$个词在第$j$个文档中的 TF-IDF，我们使用公式(1)和公式(2)计算：

$$\text{idf}(i) = \log \frac{|D|}{1+|\{t|w_i \in d_t\}|} \quad (1)$$

$$\text{tfidf}(i,j) = \text{tf}(i,j) \times \text{idf}(j) \quad (2)$$

根据 TFIDF 的值可以选取每个文档中前k个词作为该法条的特征词。在得到每个法条的特征词之后，根据特征词在待匹配文档中出现的个数为法条和待匹配文档打分，选取分数最高的法条，若分数高于阈值则匹配成功，否则匹配失败。假设第j个法条文档选取的特征词集合$w_j = \{w_{j1}, w_{j2}, \dots, w_{jk}\}$,其中$w_{j1}, w_{j2}, \dots, w_{jk}$均是词集合$W$中的词，$\text{tf}(w_t)$为待匹配文本片段上词汇$w_t$的词频。

$$score(j) = \sum_{t \in j1,j2,\dots,jk} \text{tf}(w_t) \times \text{tfidf}(t,j) \quad (3)$$

第j个法条文档的匹配度计算如公式(3)所示，通过统计集合w_j内的特征词在待匹配文本片段上的词频，可以计算第j个文档在待匹配文本片段上的得分。

2.4 特征词序列匹配

特征词序列匹配的模块针对的是文本可能出现隐式引用的情况，为了更精细地匹配文本中提及的法条。该模块起着承上启下的作用，一方面 DFA 无法识别出隐式引用的法条，而特征词序列匹配模块的输入则是根据 DFA 的识别结果筛选出来的法条，有效利用了 DFA 识别的结果分割了待匹配的文本，对待识别的片段有较强的针对性；另一方面，特征词序列匹配的结果是若干和待匹配文本片段比较符合的候选法条，候选法条成为下一个模块的输入。

如果待匹配文本引用了法条中的内容，那么它们中的某一部分应该具有较高的相似度。回顾 2.3 中选取特征词的方法，选出了不同内容的法条间具有代表性的词，但是没有考虑到特征词之间的位置信息。实际上较长的待匹配文本有更大的概率包含更多的特征词，若用此方法则可能出现多个法条匹配度都很高的情况。为了避免这种情况，我们在匹配的过程中保留了位置信息，选取特征词序列代替文本作为匹配的对象。

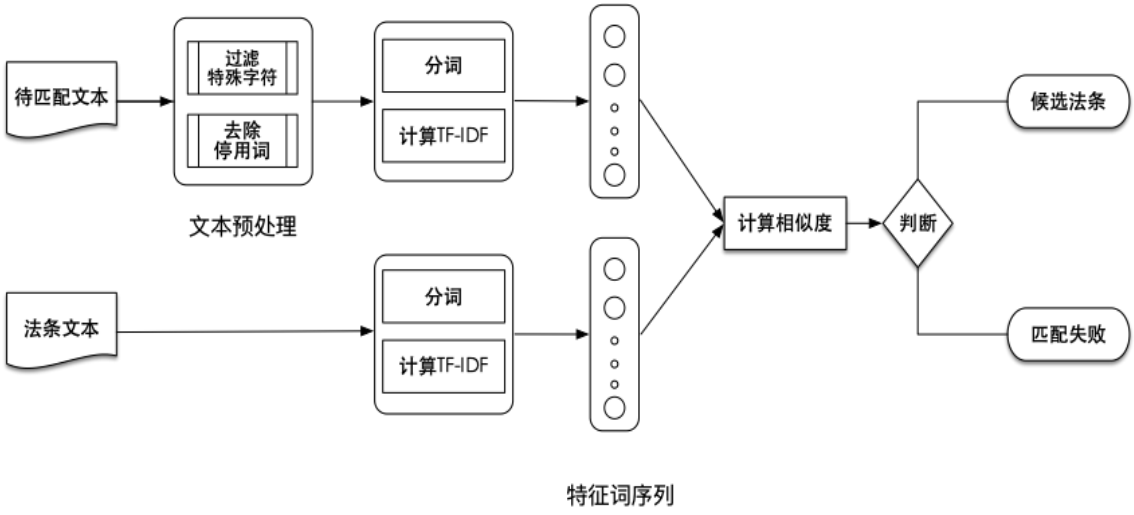

图 3 特征词序列匹配流程图

具体匹配流程如图 3 所示，在进行同样的预处理、分词和选取特征词的操作之后，计算两个序列之间得到相似度，若达到阈值则匹配成功，否则匹配失败。

假设两个特征词序列分别为 $s_1 = \{w_1, w_2, \dots, w_m\}$，$s_2 = \{t_1, t_2, \dots, t_n\}$构建匹配矩阵$\mathbf{M}_{m\times n}$，$c_{i,j}$为$\mathbf{M}$中第$i$行，第$j$列的元素，$1 \le i \le m, 1 \le j \le n$。$c_{i,j}$的值由公式(4)给出，特征词相同对应的位置元

素值为 1，不同则对应元素值为 0。

$$c_{i,j} = \begin{cases} 1, & if\ w_i = t_j \\ 0, & else \end{cases} \quad (4)$$

如果待匹配的文本中引用了法条的原文，则文本与法条的特征词序列会出现连续匹配的现象，对应到匹配矩阵上，取某一大小的方阵观察匹配矩阵，当移动到s_1和s_2相同部分的匹配位置时会发现方阵的主对角线上会出现连续的 1。为了衡量匹配程度，需要计算两个序列的相似部分之间的相似度。考虑到待匹配文本中表达不规范造成词序不严格一致的情况，允许相同的特征词在两个序列中的位置差距在 2 个词之内，因此计算相同部分的相似度时不应该只考虑主对角线上的匹配情况，也应该考虑两条次对角线上的匹配程度。且通常用逗号分隔的一句话的长度约为 8 个词，选取大小为 8 的观察窗口，在匹配矩阵上移动，根据观察窗口内主对角线和次对角线上 1 的个数来判断观察的部分序列的相似程度。

$$\text{kernel} = \begin{pmatrix} 1 & 0.5 & & & & & & \\ 0.5 & 1 & 0.5 & & & & & \\ & 0.5 & 1 & 0.5 & & & & \\ & & 0.5 & 1 & 0.5 & & & \\ & & & 0.5 & 1 & 0.5 & & \\ & & & & 0.5 & 1 & 0.5 & \\ & & & & & 0.5 & 1 & 0.5 \\ & & & & & & 0.5 & 1 \end{pmatrix} \quad (5)$$

经过实验比较最终选取大小为8×8的卷积核 kernel 如公式(5)所示，kernel 的主对角线上为 1，两条次对角线上为 0.5，使用卷积核 kernel 在匹配矩阵**M**上做卷积运算，选取卷积结果的最大值作为两个序列的匹配值，根据匹配值与卷积核大小的相对值，筛选出可能匹配的法条，淘汰相差太大的法条，将候选的法条作为深度文本匹配模型的输入，在候选的法条中进一步判断是否匹配。

如图 4 所示，待匹配文本和法条文本的特征词序列的长度可以相同也可以不同，在匹配矩阵**M**上使用卷积核 kernel 进行卷积运算，将

计算的结果使用最大池化得到两个序列之间的匹配值。

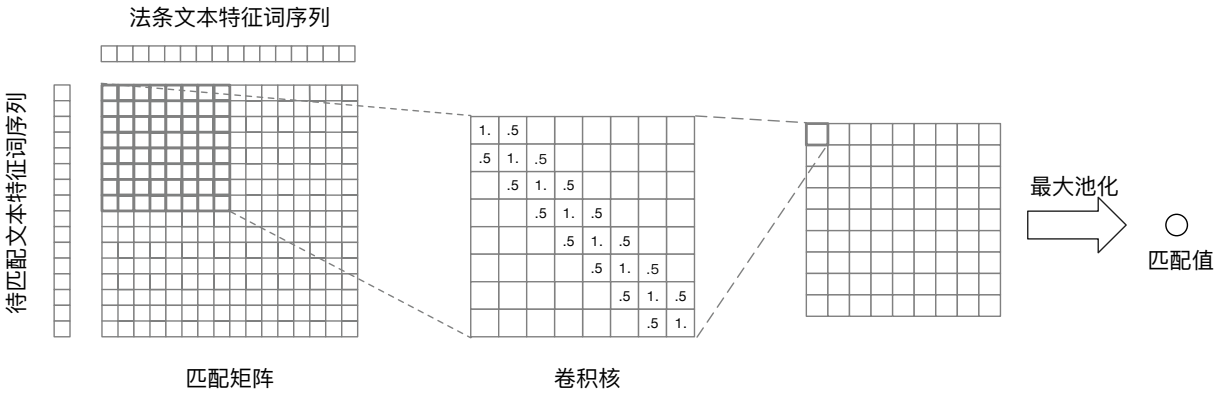

图 4 特征词序列相似度计算

$$\text{seqscore}(s_1, s_2) = \frac{p}{l} \quad (6)$$

如公式(6)所示，在实际的匹配过程中，两个序列s_1和s_2之间的相似度$\text{seqscore}(s_1, s_2)$使用匹配值$p$与 kernel 的大小$l$之间的比值表示，把序列相似度与设定的阈值$t$做比较来选取候选法条，实际过程中选取$t$的值在 0.5 左右可以获得较好的效果。

2.5 文本分段匹配

由于特征词序列匹配模块倾向于匹配上引用法律原文的情况，在含有法律原文的文本中，加入特征词序列进行匹配的 DFA 模型有很高的识别准确度，但是对于只提到法条中个别词，或者使用近义词重新表述法条内容的文本，特征词序列匹配的方法无法正确识别出文本中包含的法律和法条。为了更好地利用文本中包含的语义信息，我们考虑使用深度语义匹配模型来计算待匹配文本和法律原文的语义相似度。

由于互联网的文本数据中可能同时包含了多条法律，并且通常会把法条内容融入事件的描述或者概括法条的内容，因此如果把整段文本放入模型进行训练会引入过多的噪声，使得效果并不理想。因此，使用深度文本匹配的模型进行训练之前，需要对待匹配的文本做更加细化的处理。

在利用 DFA 模块对待匹配文本进行法律和法条的提取后，可以得到每个法律对应的文本区间。在文本包含多部法律的情况下，一般在引用法律原文之前会提到法律名，因此以法律名为切割点，分隔出每

部法律内容最可能出现的文本片段，将整段文本分割后得到文本片段的列表$P=[P_1,P_2,\dots,P_i,P_{i+1},\dots,P_n]$，其中每个片段由一个个词组成$P_i=[c_1,c_2,\dots c_m]$，每个片段$P_i$可能对应一个法条，因此训练的时候不应该把整个待匹配文本片段列表P作为输入，而应该每个P_i作为输入进行训练。

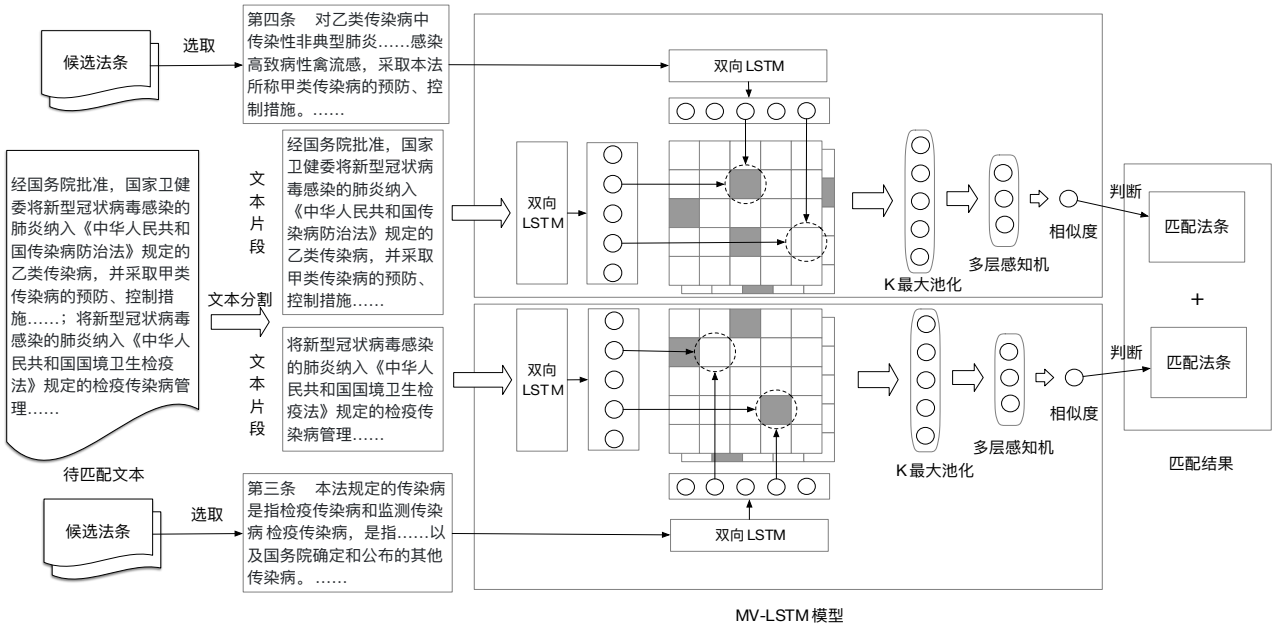

图 4 文本分段匹配示意图

我们使用的文本分段匹配模块如图 5 所示，文本分割是根据 DFA 的识别结果，对待匹配文本按照法律进行分段，再输入分割的片段计算相似度，MV-LSTM 模型基于双向 LSTM，对每一段匹配的文本正向和逆向输入 LSTM，得到两个向量表示，通过拼接两个向量得到每个位置的向量表示。然后通过张量函数对两个句子中的各个位置的向量进行交互，对得到的向量每一维进行切片得到图 5 中的交互矩阵，通过 K 最大池化每一个切片得到一个 K 维的向量 q，拼接后通过多层感知机得到一个衡量两段文本语义相似度的值。

三．结论

我们深入研究了互联网文本中法律法条的抽取问题，利用了 DFA 模型抽取文本中引用的法律和法条，使用 TF-IDF 选取特征词来区分同一部法律中不同历史版本的法条，通过计算特征词序列之间的相似度来筛选可能匹配上的候选法条，消除了数据倾斜的问题，将候选法条输入 MV-LSTM 计算得到文本与法条的语义相似度，根据语义相似度来判断文本与法条是否匹配，在法律以及法条级别匹配的F_1分别达到了 0.97 和 0.92。总的来说，本文采用的基于 DFA、特征词序列以及深度匹配的混合层次抽取模型 DS-LSTM 在法律法条引用信息抽取任务中有较好的效果。