

海量法律文书中基于 CNN 的实体关系抽取技术

高 丹, 彭敦陆, 刘 丛

(上海理工大学 光电信息与计算机工程学院, 上海 200093)
E-mail: pengdl@usst.edu.cn

摘要: 传统文本实体关系抽取算法多数是基于特征向量对单一实体对语句进行处理, 缺少考虑文本语法结构及针对多对实体关系的抽取算法. 基于此, 提出一种基于 CNN (Convolutional Neural Network) 和改进核函数的多实体关系抽取技术—KMCNN (Multi-Entity Convolutional Neural Network Based on Kernel), 并将所提技术运用于海量法律文书的实体关系抽取上. KMCNN 从抽取大规模历史法律文书的人物关系出发, 构建短语有效子树, 采用基于改进的核函数来计算短语有效子树的相似度, 以实现运用 CNN 算法对多对实体关系进行挖掘的目标. 在真实数据集上的实验表明, 所提技术具有较好的抽取效果和较高的计算效率.

关键词: 实体关系抽取; 核函数; 相似度; CNN

中图分类号: TP311

文献标识码: A

文章编号: 1000-1220(2018)05-1021-06

Entity Relation Extraction Based on CNN in Large-scale Text Data

GAO Dan, PENG Dun-lu, LIU Cong

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Most of the traditional entity relation extraction algorithms are based on feature vectors to process single-pair entities, and there are few relation extraction algorithms taking the grammatical structure and multi-entity relationship into account. This paper proposes a technology called KMCNN (Multi-Entity Convolutional Neural Network Based on Kernel) which is based on CNN (Convolutional Neural Network) and improved by a kernel function to extract the entity relationship in large-scale judicial textual documents. Based on characteristic relations extracted from the textual data, an effective phrase tree is constructed. The similarity between the trees is computed with an optimized kernel function. With the proposed approach, combined with CNN algorithm, KMCNN is designed to extract relationship among multi-entities. The effectiveness of the proposed approach is verified with the experiments which were conducted on real judicial textual documents.

Key words: entity relation extraction; kernel function; similarity; convolutional neural network

1 引言

随着计算机技术和人工智能科学发展, 使得自然语言的计算机处理成为现实. 近年来, 自然语言处理被广泛地应用到信息检索、文本分类、自动文摘、语音自动识别与合成、机器翻译及人机对话等领域. 作为自然语言理解技术中不可缺少的重要环节——文本实体关系抽取技术, 更是成为近年来的研究热点. 文本实体关系抽取是指根据自由文本的上下文, 自动抽取两个实体之间的关联. 譬如, 法律文书中句子“李良挑衅斗殴, 致韩寒休克死亡”表明两个人物实体“李良”与“韩寒”之间构成了“犯罪”关系.

自 1998 年 MUC¹ 会议首次正式提出关系抽取任务以来, 实体关系抽取已经被应用到不同的领域. 在问答系统或推荐系统中, 实体关系抽取会自动将问题、答案以及相关实体进行关联. 譬如, 当用户搜索“姚明”时, 系统会快速且准确地返回、推荐“叶莉”(夫妻关系)、“NBA”(雇佣关系). 在案由分

析系统中, 实体关系自动抽取提升了审判人员案由分析的速度, 不仅直接关系到当事人的法律关系认定, 还有利于法官对适用法律的正确选择, 形成恰当的判决结果.

迄今, 众多国内外研究学者们已经提出了一系列实体关系抽取方法. Zhou JF 等人构建抽取中文实体命名及其关系的信息抽取系统, 利用 MBL 算法获取规则以达到实体关系抽取的目的^[5]. Zhang Z 等人基于 SVM 分类器以及 bootstrapping 思想, 提出一种新的提升算法-BootProject, 实现对实体关系的半监督抽取^[6]. Sun L 和 Han X 利用特征向量提炼语法树, 基于核函数提出一种名为 FTK (Feature-Enriched Tree Kernel) 的实体关系抽取方法^[8]. 针对法律文书的半结构化、实体类型、实体之间关系单一的语言特点, 本文利用语法结构相似性构建短语有效子树, 同时采用余弦相似度计算方法来改进核函数, 求得短语有效子树之间的相似性矩阵, 然后结合 CNN 提出一种实现对多对实体之间的关系进行自动抽取的技术——KMCNN.

¹ MUC[EB/OL]. <http://www.itl.nist.gov>, 2008.

收稿日期: 2017-04-03 收修稿日期: 2017-06-20 基金项目: 国家自然科学基金项目(61003031)资助; 上海市自然科学基金项目(10ZR1421100)资助. 作者简介: 高 丹, 女, 1990 年生, 硕士研究生, 研究方向为自然语言处理; 彭敦陆, 男, 1974 年生, 博士, 教授, CCF 会员, 研究方向为大数据管理、轨迹数据压缩技术、自然语言处理; 刘 丛, 男, 1983 年生, 博士, 讲师, 研究方向为智能算法、文本挖掘、图像分析.

论文其余部分的组织如下:第2部分介绍实体关系抽取方法相关的前人研究成果;第3部分给出本文用到的术语描述及准备工作;第4部分给出基于 KMCNN 的实体关系抽取过程;第5部分采用实验对所提方法进行有效性验证;第6部分是全文的结论。

2 相关工作

过去几十年,对实体关系抽取的研究得到了人们的重视,许多实体关系抽取方法已得到广泛应用。不同模式抽取方法,如基于模式匹配^[10]的关系抽取、基于词典驱动^[11]的关系抽取、基于机器学习^[5]的关系抽取、基于 Ontology^[12]的关系抽取方法,在不同程度上推动了实体关系抽取的发展。这些方法的共同之处是将实体关系抽取任务视为分类问题。Hendrickx I 等人利用 MaxEnt、SVM 等分类器,采用特征向量完成 SemEval-2010 数据集上的实体关系自动抽取任务^[13]。Liu KB 等人开发的中文关系自动抽取系统运用改进的语义序列核函数,结合 KNN 算法构造分类器对关系类型进行分类标注^[14]。Banko M 等人通过深层解析一个相对较小的语料集,利用贝叶斯分类器进行训练以实现实体关系的抽取^[7]。

近几年来,越来越多的研究者们则将深度学习方法与 NLP 的分类任务相结合,通过深度学习的自动学习能力,对自然语言进行处理。Liu CY 等人利用同义词字典对输入词汇进行编码,将词法特征、语义知识集成到神经网络中,提出一种新的卷积神经网络挖掘实体关系^[2]。Liu K 等人利用脉冲耦合神经网络(Pulse Coupled Neural Network, PCNN)的最大池自动学习相关特性,提出一个 PCNN 与多实例学习相结合的模型^[3]。Nguyen TH 等人利用卷积神经网络的自动学习能力,通过改变滑动窗的数目,减少对外部工具、资源的依赖,实现实体关系的抽取^[4]。

无论是传统的基于特征量及核函数的实体关系抽取方法,还是近年来兴起的基于深度学习的实体关系抽取方法,均基于仅包含单对目标实体对语句的特定数据集,提高了对原始数据进行预处理的难度。本文试图在包含多对实体的语句中完成实体关系抽取的任务,并以大规模法律文书数据中进行实体关系抽取为例进行说明。具体的算法思想如下:利用中文语法结构的局部相似性,构建短语有效子树挖掘模型,并采用基于改进的核函数来计算子树之间的相似度。基于此,提出基于 CNN 算法的多实体关系抽取方法——KMCNN,最后通过实验来验证所提算法的有效性。

3 准备工作

本节主要介绍如何对原始文本进行预处理,以适合所提算法的计算要求。在给出下文所需相关术语的基础上,提出详细的短语有效子树挖掘过程,然后采用改进的核函数来计算短语有效子树相似度。

3.1 术语解释

实体(Entity):自由文本中具有特殊含义的概念,记为 e 。

实体关系(Relation):一对实体间具有的联系,记为 $R(e_i, e_j, r_{i,j})$,其中 (e_i, e_j) 为实体对, $r_{i,j}$ 为实体对 (e_i, e_j) 之间的关系。

例如,在法律文书的案情描述中:人物实体为施害人、被害人;实体之间的关系则为死亡或重伤等犯罪事实。

表1中给出了下文将要用到的符号及其所表示的含义。

表1 算法中所用到的符号说明

Table 1 Explanation of words used in paper

名称	含义
W	词序列, $W = \{w_1, w_2, \dots, w_n\}$
E	词向量矩阵, $E = \{e_1, e_2, \dots, e_n\}$
D	距离向量矩阵, $D = \{d_1, d_2, \dots, d_n\}$
P	句子中短语集合, $P = \{p_1, p_2, \dots, p_m\}$
T'	短语有效子树集合 $T' = \{T'_1, T'_2, \dots, T'_n\}$
A	短语有效子树的相似性矩阵
S	短语有效子树的节点序列
$Matrix$	向量全矩阵 $Matrix = \{m_1, m_2, \dots, m_n\}$
ε	有效子树节点相似性阈值
win	滑动窗口数目
α	卷积核大小

3.2 短语有效子树挖掘

从中文语法结构出发,短语是句子的主要成分。短语结构树被视为句子语法结构的可视化,可用于挖掘句子中的隐藏信息。

定义1. 短语:由若干个连续的词序列 $w_i \sim w_{i+n_s}$ 搭配成的独立语言单位,记为 P_i 。对于一个给定的词序列 W ,将其分割成若干短语集合的过程,记为 $P = \{P_i | 1 \leq i \leq n, n \text{ 为句子的词组个数}\}$ 。

例如,对词序列“李良挑衅斗殴,致韩寒休克死亡”进行短语切割,得到短语集合 $P = \{\text{李良, 挑衅斗殴, 致, 韩寒, 休克死亡}\}$,其中,“挑衅斗殴”等词即为短语。

定义2. 短语有效子树:给定一棵有序的语法树 $T = (V, E, R)$,其中, V 表示节点集合, E 表示所有的路径集合, R 是根节点。当 $T' = (V', E', R')$ 满足:

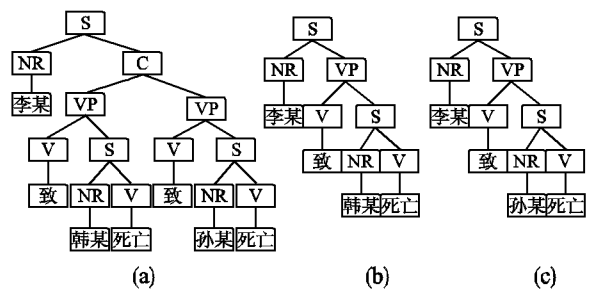


图1 短语有效子树

Fig. 1 Effective subtree of phrases

- 1) $V' \subseteq V, E' \subseteq E$;
- 2) V' 包含树 T 中 R' 的所有子孙结点
- 3) T' 中的节点序列由若干 P_i 构成,且有且仅有两个 NR 节点。则 T' 称为 T 的短语有效子树。如图1所示,树 b、c 为语法树 a 的两棵短语有效子树。在短语有效子树的挖掘算法中应用以下规则:

规则1. 子树根节点挖掘规则

语法树中的每个节点符号均有可能在每一个实例中出

现,如图 1 中的节点 S。若对其进行特征选择,则多数子树都会是无效的,以致产生错误的结果,因此挖掘子树根节点(即单项目集)时,所有非叶节点的单项均是有效的。

短语库 $H \langle Type, Num \rangle$ 为所有短语类型的集合, $Type$ 为短语类型, Num 为该短语中词汇的数目。因此,子树根节点集 $Trie = \{t_i \in H \cap T'. \forall 1 \leq i \leq n, n \text{ 为有效子树个数}\}$

规则 2. 投影序列片段产生规则

自左至右,先序遍历语法树,则节点内容、节点序号信息加入树序列。以图 1 为例,以 S 为根节点,则该子树的序列为 S1-NR2-VP3-V4-S5-NR6-V7(李某致韩某死亡)。

挖掘短语有效子树的具体过程如下:依次遍历子树中的序列,并判断该序列的节点类型,若节点类型存在于短语库中,则该节点是一棵短语有效子树的根节点,获得以该节点为根节点的所有短语有效子树全序列。算法 1(图 2)详细描述了该过程。算法前 4 行完成数据的初始化,第 1 行初始化短语有效子树为空集。第 2 行利用 $transfer()$ 函数将语法树转换成二叉树,便于之后的遍历操作,第 3 行 $preOrder()$ 函数前序遍历该二叉树,并转换成投影序列 S,第 4 行 $length()$ 函数取得序列的长度。5-15 行依次遍历序列,挖掘短语有效子树。第 6 行 $isContain()$ 函数判断序列节点是否存在于短语库中,若存在,则跳入 7-12 行,其中 7-9 行利用 $preAppend()$ 函数在该子树前端节点依次插入序列节点,10-12 表示将该子树的最后节点替换成相对应的子树。最后,15 行根据定义 3 移除无效短语子树,并返回。

3.3 基于核函数的相似度计算

目前,通过核函数计算相同子树的个数是计算两棵树的相似度的经典方法。但该方法忽略了子树结构,隐藏了文本隐含信息,不利于实体关系抽取的准确度。基于此,论文提出改进的核函数,计算两棵短语有效子树的相似度。

定义 3. 相似矩阵:由短语有效子树的个数确定矩阵的维数,矩阵元素 $A[i, j]$ 表示 T'_i 与 T'_j 的相似程度。若短语有效子树完全相同,则 $A[i, j] = 1$, 反之 $A[i, j] = 0$ 。

两棵短语有效子树的相似度计算是对节点类型、短语语法结构相似程度的度量,其求解过程的主要步骤就是构建相似矩阵。去掉短语有效子树的叶子节点,应用余弦相似度计算有效子树对应节点的相似度:

$$\cos \theta_{ij} = \frac{\cos \theta_{ij} \sum_1^d (A_i \times B_j)}{\sqrt{\sum_1^d A_i^2} \cdot \sqrt{\sum_1^d B_j^2}} \quad (1)$$

其中,向量 $A_i \in R^{d \times n_w}$, $B_j \in R^{d \times n_w}$ 是两棵子树的所有节点构成的向量。基于上述计算,当相似度大于某个阈值时,两个节点近似相同。这样,就可以得到核函数:

$$K(T'_i, T'_j) = \emptyset(T'_i) \cdot \emptyset(T'_j) = \sum_{n_1 \in T'_i} \sum_{n_2 \in T'_j} \Delta(n_1, n_2) \quad (2)$$

其中,

$$\Delta(n_1, n_2) = \begin{cases} \prod_{j=1}^{nc(n_1)} (1 + \Delta(c(n_1, j), c(n_2, j))) \cos \theta_{ij} > \varepsilon \\ 0 & \text{其他情况} \end{cases} \quad (3)$$

算法 2(图 3)详细的描述了如何求解相似矩阵。算法第 1 行 $size()$ 函数计算有效子树集合的子树个数。第 2 行通过 $zero()$ 函数初始化相似矩阵 A 为 0 矩阵,表示子树两两均不相

似。3-13 行完成相似矩阵的求解,其中第 6 行根据公式 2 求解两棵子树之间的相似度,若大于 ε ,则 7-9 行设置对应的相似矩阵元素为 1。最后,13 行返回相似矩阵。

算法 1. 挖掘短语有效子树

输入:语法树 T, 短语库 $H \langle Type, Num \rangle$

输出:短语有效子树集合 T'

BEGIN;

1 $T' = \emptyset$;

2 $transfer(T, T_{mind})$://将 T 转换成二叉树 T_{mind}

3 $S = preOrder(T_{mind})$ //前序遍历 T_{mind} , 生成 S

4 $len = length(S)$;

5 FOR i in len:1//倒序遍历 S 的每个序列

6 IF($isContain(H, Type, S_i)$)// S_i 是 H, Type 的子集

7 FOR j in 1:H.num//循环获取 S_i 的前 H.num 个序列

8 $T'_i.preAppend(S_{i+j})$;

9 END FOR;

10 IF($isContain(H, Type, T'_i.lastnode)$)//若 T'_i 的最后一个节点 $lastnode$ 是 H, Type 的子集,则用相应的子树替换

11 $T'_i.replace(lastnode, T'_{lastnode})$;//替换 $lastnode$

12 END IF;

13 END IF;

14 END FOR;

15 RETURN($T'.remove()$);

END

图 2 短语有效子树挖掘算法 PSTMining

Fig. 2 Algorithm of PSTMining

算法 2. 计算相似矩阵

输入:短语有效子树集合 T' , 相似度阈值 ε

输出:相似性矩阵 A

BEGIN;

1 $size = size(T')$;

2 $A = ZERO(SIZE, SIZE)$;

3 FOR i in 1:size//遍历 T' , 针对 T' 中的每一棵短语子树 T'_i

4 FOR j in i:size//遍历 T'_i 之后的每棵短语子树 T'_j

5 IF($len(T'_i) = len(T'_j)$)//若 T'_i 与 T'_j 的序列长度相同

6 $Sim(i, j) = K(T'_i, T'_j)$;//计算相似度的值

7 IF($Sim(i, j) \geq \varepsilon$)//若相似度大于相似度阈值

8 $A[i, j] = A[j, i] = 1$;//设置矩阵元素为 1

9 END IF;

10 END IF;

11 END FOR;

12 END FOR;

13 RETURN(A);

END

图 3 求解相似矩阵算法 SimMatrix

Fig. 3 Algorithm of SimMatrix

4 实体关系抽取技术—KMCNN

通过卷积神经网络(CNN)的自动学习能力,可以减少构建大规模语料库的人力耗费,实现多实体关系的自动抽取。在前文本数据预处理的基础上,本节将重点讨论 KMCNN 模型。

4.1 相关概念

自然语言处理过程中的主要任务是如何对词、句子、篇章进行编码,以便将其作为数值类型的数据输入到模型中进行

计算。

定义 4. 词向量 (Word Vector): 词序列中的每个“词”均可表示成一个 d 维实数向量 $e_i \in R^d, i = 1, 2, \dots, n$ 。

定义 5. 距离向量 (Distant Vector): 词 e_i 与两个实体之间的距离向量, 记为 $dist_i = \{ (dist_{i_1}, dist_{i_2}), i = 1, 2, \dots, n \}$ 。即为短语有效子树节点之间的边数。

定义 6. 向量全矩阵 (The Full Embedding Matrix): $Matrix = [m_1, m_2, \dots, m_n] \in R^{(d+2) \times n}$, 其中, n 是词序列的长度。对于一个给定的词序列 $W = \{w_1, w_2, \dots, w_n\}$, 词向量 v_i 是第 i 个词 w_i 对应的一个由词向量 e_i 与距离向量 $dist_i$ 组成的 $d+2$ 维实数向量, 即 $m_i = [e_i, dist_i]$ 。

4.2 基于相似性矩阵求解向量全矩阵集合

传统的基于 CNN 的实体关系抽取算法多数是针对单对实体的, 而法律文书中包含实体的句子通常包含多对实体, 并且语义结构具有相似性。针对这一发现, 利用短语有效子树的相似性矩阵对句子进行切分, 并假设: 同一短语中出现多个实体、当两棵短语有效子树的相似性值大于阈值时两个短语中的所有实体均为并列关系, 即同时成为施害人或被害人。

算法 3. 基于相似矩阵求解向量全矩阵

输入: 短语有效子树集合 T' 相似性矩阵 A

输出: 向量全矩阵集合 $Matrix = \{len, set = \{m_1, m_2, \dots, k_{len}\}\}$

BEGIN;

1 $size = size(A)$; // 计算 A 的维数, 即 T' 的短语子树的个数

2 $Matrix = \emptyset$;

3 FOR i in $1:size$

4 $m_i.len = 0$;

5 FOR j in $i:size$

6 IF $(A[i, j] = 1)$ // 若矩阵元素为 1

7 $E_i = Word2Vec(seq(T'_i))$; // T' 转化成词向量

8 $D_i = dist(E_i)$; // 求解 E_i 的距离向量

9 $m'_i = m_i$; // 利用中间距离 m'_i 保存 m_i

10 $m_i = \{m'_i, len + 1, m'_i.set.append([E_i, D_i])\}$;

11 END IF;

12 END FOR;

13 END FOR;

14 RETURN($Matrix$);

END

图 4 求解向量全矩阵 EntireMatrix

Fig. 4 Algorithm of EntireMatrix

基于相似矩阵, 算法 3 (图 4) 详细描述了如何求解 KMCNN 中的向量全矩阵参数: 有效子树集合与向量全矩阵集合。第 1-2 行对数据进行初始化, 向量全矩阵集合 $Matrix$ 为空集, 其中 len 参数记录该集合的长度。核心代码为 3-13 行: 第 4 行初始化 len 为 0, 表示当前集合中全矩阵数目为 0; 第 6 行判断相似矩阵元素的值, 若值为 1 则跳至第 7 行, 应用 $Word2Vec^{[15]}$ 将短语有效子树的序列化数据转换成词向量; 然后, 第 8 行 $dist()$ 函数求解相对应的距离矩阵; 第 10 行更新向量全矩阵集合, 即 $append()$ 函数将向量权矩阵在添加至集合中, 同时设置集合长度加 1。最后, 14 行返回结果。

4.3 KMCNN

前文介绍了如何挖掘短语有效子树, 并基于改进的核函数对相似性矩阵进行计算, 求得向量全矩阵。下面介绍基于

KMCNN 来实现实体关系抽取的过程。图 5 给出了 KMCNN 的伪代码: 代码第 1 行将实体关系集初始化为空集, 并设索引值为 0。第 2-11 行遍历向量全矩阵集合, 依次抽取实体对之间的关系。其中, 3-10 行完成指定集合的实体关系抽取: 首先, 第 4 行基于分词、词性标注等知识, 应用 $reconge()$ 函数对实体进行识别; 然后, 判断实例是否为集合中的第一个元素, 若是, 则跳至第 6 行, 基于 CNN 算法对实体关系进行抽取; 最后, 第 12 行返回实体关系集合。

算法 4. 多实体关系自动抽取方法

输入: 向量全矩阵集合 $Matrix, m$

输出: 实体关系集合 R

BEGIN;

1 $R = \emptyset, index = 0$;

2 FOR m_i in $Matrix$ // 遍历 $Matrix$, 针对 $Matrix$ 的每一个向量全矩阵 m_i

3 FOR i in $1:m_i.set.len$ // 遍历 m_i , 针对 m_i 的每一个矩阵 $m_i.set[i]$

4 $entitys = reconge(m_i.set[i])$; // 抽取 $m_i.set[i]$ 的实体 $entitys$

5 IF $i = 1$ // 若当前矩阵是首个向量全矩阵

6 $r = cnn(m_i.set[i])$; // 利用 cnn 抽取实体关系

7 END IF;

8 $index++$; // 否则, 索引加 1

9 $R_{index} = [entitys, r]$; // 存储实体及实体关系

10 END FOR;

11 END FOR;

12 RETURN(R);

END

图 5 KMCNN 方法

Fig. 5 Approach of KMCNN

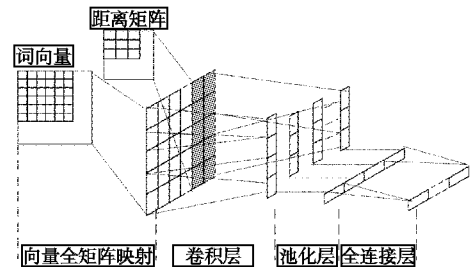


图 6 CNN 算法结构图

Fig. 6 Structure of convolutional neural network

如图 6 所示, CNN 算法包含四个主要部分: 全向量映射、卷积层、池化层以及全链接层。其中, 全向量映射是根据定义 7 求解全向量矩阵的过程。在卷积层, 若滑动窗口数目为 win , 则卷积核的权重集合是:

$$f = \{f_1, f_2, \dots, f_{win} | f_i \in R^{(d+2) \times n}\} \quad (4)$$

基于公式 (4), 给出卷积值的计算公式:

$$C = [c_1, c_2, \dots, c_{n-win+1} | c_i = g(\sum_{j=0}^{win-1} f_{j+1}^T m_{j+i}^T + b)] \quad (5)$$

其中, b 为偏置值, g 是一个非线性函数。然后, 在池化层运用最大池化原理提取最大卷积值, 即 $p_{max} = \max(C)$ 。最后, 在全连接层采用 $sigmoid$ 函数实现实体关系的抽取。

5 实验分析

5.1 数据来源

实验部分的数据采集于 2016 年某省刑事案件的律文

书²,共 25,463 份文本数据。裁定书的内容主要包含以下五部分:被告人信息;以时间为序,开庭判决过程;复核事实;证据陈述;判决结果。因此,可将裁判文书视为有模板的半结构化数据,利用正则表达式匹配全文信息来提取关键段落(即被告人信息、复核事实等),并进行数据预处理过程(去除噪声数据、重复数据、提取包含实体对的句子)。

5.2 算法有效性分析

准确率(Precision)、召回率(Recall)和 F1-measure 是评估算法有效性的基本标准。因此,实验采用三个指标对所提算法进行综合性评估。下面,给出三个指标的数学定义:

$$Precision = \frac{N_c}{N_c + N_{ic}} \quad (6)$$

$$Recall = \frac{N_c}{N_{sum}} \quad (7)$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (8)$$

N_c 是某类别中被正确分类的实例数目, N_{ic} 是某类别中被错误分类的实例数目, N_{sum} 是某类别中的实例总数。

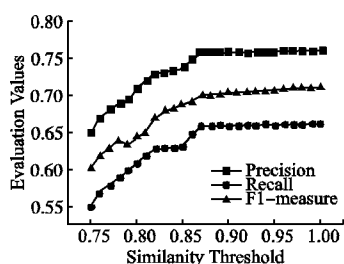


图 7 不同相似度阈值下的实体关系抽取结果

Fig. 7 Relation extraction in KMCNN vs. similarity

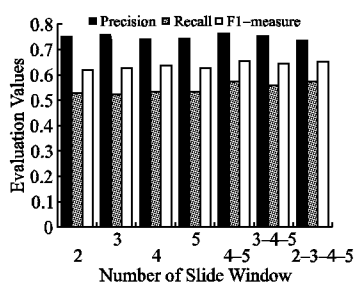


图 8 不同滑动窗口数目下的实体关系抽取结果

Fig. 8 Relation extraction in KMCNN vs. slide window number

第 1 组. 在真实数据集上实现实体关系的抽取

实验 1 考察了在不同相似度阈值下,KMCNN 的实体关系的抽取效果。KMCNN 需要提供相似度阈值 ε 确定短语有效子树之间的相似性矩阵。实验中发现不同的 ε 取值,对于最终的实体关系抽取结果有很大的影响。图 7 是实验结果,横轴表示相似度阈值,纵轴表示三个指标的值。从图中可以看出,当 ε 从 0.75 到 0.87 变化时,三个指标值上升很快,表明实体关系抽取效果越来越好。而当 ε 大于 0.87 后,三个指标值趋于平稳,即实体关系抽取效果趋于稳定。

实验 2 考察不同滑动窗口数目下,KMCNN 的实体关系

的抽取效果。实验中滑动窗口的数目包含两种:固定滑动窗口大小,取值分别是 2、3、4、5;组合滑动窗口大小,组合取值分别是(2,3,4,5)、(3,4,5)、(4,5)。图 8 中显示了滑动窗口数目对实体关系抽取结果的影响:(1)滑动窗口数目固定时,KMCNN 的抽取效果不稳定。滑动窗口大小为 3 时,准确率较高;滑动窗口大小为 2、5 时,召回效果较好。(2)组合滑动窗口大小的取值时,KMCNN 的抽取效果稳定并呈现较好的趋势。特别地,滑动窗口组合大小(4,5)时,抽取结果具有很高的准确率,且召回效果良好。

第 2 组. 考查 KMCNN 计算性能

在确定了相似度阈值和滑动窗口数目的基础上,本组实验将验证相似性取值为 0.87、滑动窗口数目为(4,5)时,使用所提算法进行实体关系抽取的计算效果。本文从两个方面对 KMCNN 与 O-CNN^[1]、W-ONN^[4]、MVRNN^[8]等现有算法的实体关系抽取结果进行考察:

实验 3 考察 KMCNN 的运行效率。图 9 中显示了四种算法在不同数据集规模的情况下,抽取实体关系所需要的运行时间。在数据集规模小于 10,000 时,相对于其余三种算法,KMCNN 在较短的时间内完成实体关系的抽取。随着数据规模的增加,四个算法的运行时间的差距增大。这是因为随着数据集规模的增加,挖掘短语有效子树的时间明显减少,意味着 KMCNN 算法的计算规模也明显减少。

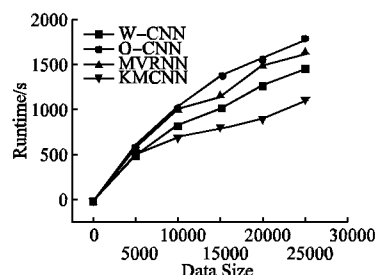


图 9 不同数据集规模下算法的运行时间比较

Fig. 9 Different algorithms runtime vs. data set

实验 4 验证 KMCNN 的实体关系抽取结果的准确性。图 10 中显示了 4 个算法在不同的数据集规模(分别选取 5000, 10000, 15000, 20000 篇法律文书)下得到的实体关系抽取结果的三个指标值。文献[1, 4, 8]显示, O-CNN、W-ONN、MVRNN 三种算法都能够较好地抽取实体之间的关系,而 KMCNN 的实体关系抽取结果与 W-CNN 的实体抽取结果近乎相同,且明显优于 O-CNN、MVRNN 的实体关系抽取结果。由此可见 KMCNN 能够较好地抽取实体之间的关系。

6 结 论

实体关系抽取是自然语言处理的重要任务。快速而准确地抽取实体间的关系,对自由文本信息挖掘、主题挖掘、问答系统、推荐系统均具有重要意义。本文提出一种基于改进核函数和 CNN 的多实体关系抽取技术—KMCNN。算法利用语法结构相似性挖掘短语有效子树,通过余弦相似度计算来改进

² China Judgements Online. <http://wenshu.court.gov.cn>, 2016.

核函数,并利用该核函数计算关系实例间的相似度,结合 CNN 算法对实体关系进行抽取.算法合理运用了语法结构,

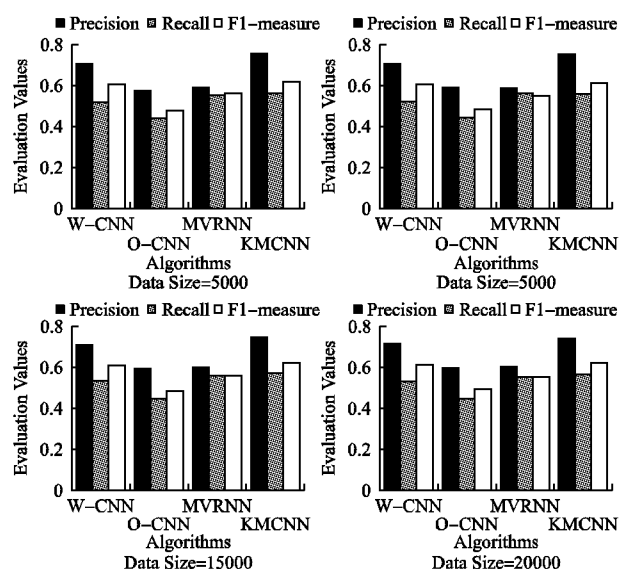


图 10 不同数据集规模下的实体关系抽取结果比较

Fig. 10 Relation extraction in different algorithms vs. data set

结合 CNN 算法的自动训练能力,不需要大规模语料库为基础,较大地减少了中间特征向量的计算量同时挖掘了句、篇中隐含的有效信息.实验结果表明,KMCNN 具有较好的实体关系抽取效果,在效率方面也有较大提高.下一步工作将围绕如何进一步提高算法效率、构建实体关系图谱及采用 MapReduce 进行分布式计算等问题展开研究.

References:

- [1] Zeng D, Liu K, Lai S, et al. Relation classification via convolutional deep neural network [J]. In Proceedings of COLING, the 25th International Conference on Computational Linguistics, 2014: 2335-2344.
- [2] Liu C Y, Sun W B, Chao W H, et al. Convolutionneural network for relation extraction [M]. Advanced Data Mining and Applications, 2013: 231-242.
- [3] Zeng D, Liu K, Chen Y, et al. Distant supervision for relation extraction via piecewise convolutional neural networks [C]. Conference on Empirical Methods in Natural Language Processing, 2015: 1753-1762.
- [4] Nguyen T H, Grishman R. Relation extraction: perspective from convolutional neural networks [C]. The Workshop on Vector Space Modeling for Natural Language Processing, 2015: 39-48.
- [5] Zhang Y, Zhou J F. A trainable method for extracting Chinese entity names and their relations [C]. The Workshop on Chinese Language Processing: Held in Conjunction with the Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2000: 66-72.
- [6] Zhang Z. Weakly-supervised relation classification for information extraction [C]. ACM CIKM International Conference on Information and Knowledge Management, Washington, Dc, Usa, November, DBLP, 2004: 581-588.
- [7] Banko M, Cafarella M J, Soderland S, et al. Open information extraction from the web [C]. International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers Inc, 2007: 2670-2676.
- [8] Sun L, Han X. A feature-enriched tree kernel for relation extraction [C]. Meeting of the Association for Computational Linguistics, 2014: 61-67.
- [9] Socher R, Huval B, Manning C D, et al. Semantic compositionality through recursive matrix-vector spaces [C]. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2012: 1201-1211.
- [10] Appelt D E, Hobbs Jr, Bear J, et al. SRI international FASTUS system: MUC-6 test results and analysis [C]. In Proceedings of the 6th Message Understanding Conference (MUC-6), 1995: 237-248.
- [11] Aone C, Ramos Santacruz M. REES: a large-scale relation and event extraction systems [C]. In Proceedings of the 6th Applied Natural Language Processing Conference, New York, 2000: 76-83.
- [12] Iria J. T-Rex: a flexible relation extraction framework [C]. In Proceedings of the 8th Annual Colloquium for the UK Special Interest Group for Computational Linguistics, 2005.
- [13] Hendrickx I, Kim S N, Kozareva Z, et al. SemEval-2010 task 8: multi-way classification of semantic relations between pairs of nominals [C]. The Workshop on Semantic Evaluations: Recent Achievements and Future Directions, Association for Computational Linguistics, 2009: 94-99.
- [14] Liu Ke-bin, Li Fang, Liu Lei, et al. Implementation of a kernel-based Chinese relation extraction system [J]. Journal of Computer Research and Development, 2007, 44(8): 1406-1411.
- [15] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [J]. Computer Science, 2013.

附中文参考文献:

- [14] 刘克彬, 李芳, 刘磊, 等. 基于核函数中文关系自动抽取系统的实现 [J]. 计算机研究与发展, 2007, 44(8): 1406-1411.