计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.191200183

基于 Zipf's 共生矩阵分解的开放域事件向量计算方法

高李政[1]　周　刚[1]　黄永忠[2]　罗军勇[1]　王树伟[1]
1 数学工程与先进计算国家重点实验室　郑州 450001
2 桂林电子科技大学计算机与信息安全学院　广西 桂林 541000
(gao1440429064@foxmail.com)

摘　要　事件抽取是自然语言处理(Natural Language Processing,NLP)领域的一个研究热点。现有的事件抽取模型大多基于小规模训练集,无法应用于大规模开放领域。针对大规模开放域事件抽取中事件表征困难的问题,提出了一种基于 Zipf's 共生矩阵分解的事件向量计算方法。首先,从开放语料中提取事件元组作为事件标签,并对事件元组进行抽象、剪枝和消歧。然后,利用 Zipf's 共生矩阵表示事件的上下文分布,利用主成分分析(Principal Component Analysis,PCA)对共生矩阵进行分解,得到初始事件向量,并利用自编码器对初始事件向量进行非线性变换。采用最近邻检测和事件检测两种任务对事件向量的性能进行测试,结果表明,基于 Zipf's 共生矩阵分解得到的事件向量能够对事件之间的相似性和相关性信息进行全局性表征,避免编码过细而造成语义偏移。

关键词:开放域事件抽取;Zipf's 共生矩阵;上下文分布;事件表征
中图法分类号　TP391

Open Domain Event Vector Algorithm Based on Zipf's Co-occurrence Matrix Factorization

GAO Li-zheng[1],ZHOU Gang[1],HUANG Yong-zhong[2],LUO Jun-yong[1] and WANG Shu-wei[1]
1 State Key Laboratory of Mathematical Engineering and Advanced Computing,Zhengzhou 450001,China
2 School of Computer Science and Information Security,Guilin University of Electronic Technology,Guilin,Guangxi 541000,China

Abstract　Event extraction is one of the hot topics of natural language processing (NLP). Existing event extraction models are mostly trained on small-scale corpora and are unable to be applied to open domain event extraction. To alleviate the difficulty of event representation in large-scale open domain event extraction,we propose a method for event embedding based on Zipf's co-occurrence matrix factorization. We firstly extract event tuples from large-scale open domain corpora and then proceed with tuple abstraction,pruning and disambiguation. We use Zipf's co-occurrence matrix to represent the context distribution of events. The built co-occurrence matrix is then factorized by principal component analysis (PCA) to generate event vectors. Finally,we construct an autoencoder to transform the vectors nonlinearly. We test the generated vectors on the task of nearest neighbors and event identification. The experimental results prove that our method can capture the information of event similarity and relativity globally and avoids the semantic deviation caused by the too fine granularity of encoding.

Keywords　Open domain event extraction,Zipf's co-occurrence matrix,Context distribution,Event representation

1　引言

事件是一种包含时间、地点、参与者、属性等要素的复杂结构,不同事件类型往往对应不同结构。由于事件种类繁多且结构复杂,自动地进行事件标注非常困难,人工标注同样费时费力,因此事件抽取数据集的规模通常较小。例如,ACE 2005 是事件抽取任务最常用的数据集,该数据集仅定义了 33 种事件类型,包含 599 个文档和大约 6 000 个标记语句,大约有 60%的事件类型的标记样例少于 100 个[1]。而另一种标注数据集——TAC KBP 2015,也只包含 38 种事件类型和 562 个文档。

随着神经网络在 NLP 领域的广泛应用,研究人员开始将神经网络模型应用到事件抽取任务中。各种各样的复杂模型,提升了事件抽取在小规模数据集上的测试效果。尽管如此,研究人员依然忽视了一个重要的问题,即当模型的参数数量和结构复杂度不断提升时,训练数据的规模也应相应增加,否则模型极易出现过拟合。此外,由于现有的标注数据集包含较少的事件类型,训练的模型无法应用于大规模开放域事件抽取。本文关注大规模开放域事件抽取问题,由于事件向量计算是事件抽取任务的基础,本文将重点研究大规模开放域条件下的事件向量表征。

词向量旨在生成词语的向量表征,并利用向量的空间相

到稿日期:2019-12-31　返修日期:2020-04-28　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61602508,61866008)
This work was supported by the National Natural Science Foundation of China (61602508,61866008).
通信作者:周刚(gzhougzhou@126.com)

邻性表示词语的语义相似性。类似地，事件向量旨在计算不同事件类型的向量表征。向量的相邻性，反映了事件类型之间的相似或相关关系。由于词汇表中的词形数量有限，在计算词向量时通常使用词形来作为词汇标签。与之对应，事件通常由语句来表示，语句是不同词汇的组合，数量极为庞大。直接将语句作为事件标签会导致“维度灾难”，使得事件样例极为稀疏。因此，本文首先从事件句中提取出能够表示事件类型的关键信息来组成事件元组，并将其作为事件类型标签。

现有的词向量计算方法大多基于 Harris 的分布假说[2]。该假说认为：如果两个词语拥有相似的上下文分布，它们的语义通常也相似。受此启发，本文认为拥有相似上下文分布的事件之间也存在相似性，因此借鉴词向量计算的思想来构建事件向量生成模型。其中，词向量模型采用的是文献[3]提出的一种基于 Zipf's 共生矩阵分解的词向量计算方法。该方法得到的词向量在多个任务中取得了与基于神经网络的词向量(Word2Vec[4]，GloVe[5]和 FastText[6]等)相近的性能，且与之相比大大缩短了计算时间。该方法直接将语句作为上下文，因此能够方便地移植到事件向量计算任务中。实验结果表明，利用本文方法得到的事件向量包含了事件的相似性和相关性信息，这些信息能够在后续的事件关系网络构建中发挥重要作用。本文得到的事件向量在事件检测任务中也显示了良好的性能。

2 相关工作

2.1 事件抽取

事件抽取是信息抽取的重要分支，包含两个主要内容：事件检测和论元提取。最初的事件抽取方法大多基于模式匹配，这些方法在特定领域能够取得较好的抽取效果，但跨领域和可移植性较差，并且需要人工提取事件模式，费时费力。随后，研究人员尝试结合特征工程和监督学习构建事件抽取模型。这些方法可以在一定程度上减少领域专家的参与度，但仍需要人工分析事件特征。近年来，随着神经网络和词向量技术在 NLP 领域的广泛应用，研究人员开始利用神经网络构建事件抽取模型，并将词向量作为模型的初始特征，避免了复杂的特征工程。基于神经网络的事件抽取方法取得了当前最好的测试效果，下文主要对这类方法进行介绍。

文献[7]基于 CNN 构建事件检测模型。该方法将预训练的词向量、位置向量以及实体类型向量级联后作为初始特征，利用 CNN 对特征进行进一步的抽象和加工，减少了特征工程带来的错误传播，提高了事件检测的性能。

Max-Pooling 层只提取句子中最重要的信息，忽视了句子级的事件关系信息。文献[8]提出了一种 DMCNN(Dynamic Multi-pooling Convolutional Neural Network)模型。该模型利用触发词和论元将事件句划分为多个部分，然后对每个部分分别进行池化，这样能够提取语句中长距离的依赖信息。

传统 CNN 模型难以捕捉语句中的句法依赖关系。针对这一问题，文献[9]提出的利用 GCN 对事件句的句法依存树进行卷积。由于 RNN 比 CNN 更适合提取上下文信息，文献[9]模型在进行卷积之前首先利用双向 LSTM 对句子进行编码。

文献[10]提出了一种 dbRNN 模型，该模型在双向 LSTM 的基础上增加了依赖边和依赖门，能够同时提取句子中的语法依存信息和时序信息。

文献[11]提出了一种 DAG-GRU 模型。该模型与 dbRNN 相似，即利用 RNN 将语句中的依存关系和时序信息进行整合；不同点在于，DAG-GRU 不需要额外增加门结构，而是利用注意力机制控制依存信息所占的比重。

事件抽取包含多个子任务。传统的事件抽取方法通常依次进行，难以综合利用各子任务之间的关联关系。文献[12]提出了一种基于神经网络的状态转移模型。该方法对实体、触发词、论元等进行联合预测，测试结果优于传统的顺序型方法。文献[13]对不同的子任务训练共享的隐藏层，并对损失函数进行联合优化，得到的模型能够同时处理实体检测、事件检测、论元角色识别等多个子任务。

文献[14]提出了一种基于对抗模仿的知识蒸馏方法。该方法首先利用标注数据训练教师模块，得到知识表征；然后利用无标注的纯文本训练学生模块。在对抗判别器的指导下，学生模块对教师模块进行模仿学习。最终得到的学生模块能够用于事件检测任务。

基于神经网络的事件抽取模型取得了当前最好的测试效果。然而，这些模型的训练和测试都是在小规模数据集上进行的，很难应用到大规模开放域事件抽取任务中。

2.2 事件表征

事件表征是事件抽取的前提，事件抽取技术的发展很大程度上归功于事件表征技术的不断改进。如上文所述，研究人员最初使用事件模式表征事件；随后利用特征工程提取事件特征来表征事件。随着神经网络模型和词向量技术在 NLP 领域的成功应用，研究人员在词向量的基础上利用各种神经网络模型计算事件的向量表征。事件的向量表征被称为事件向量。基于神经网络的事件抽取模型的大部分工作都是在生成事件的向量表征。受限于训练数据的规模，这些事件向量无法应用于开放领域。

为了能够从任意语料中提取事件，文献[15]提出了一种面向开放域的事件向量计算方法。该方法利用 AMR(Abstract Meaning Representation)提取候选触发词、论元及其依赖关系，得到事件结构；结合 WSD(Word Sense Disambiguation)和词向量技术生成触发词语义向量；利用 TRAE(Tensor Based Recursive Autoencoder)对事件结构进行处理，生成事件向量。文献[16]提出了一种基于神经张量网络模型(Neural Tensor Network)的事件向量计算方法。该方法首先利用 NLP 工具提取候选触发词、论元、语法关系等，然后利用张量运算对这些信息进行编码。文献[17]对文献[16]提出的方法进行了改进，在计算事件向量时加入了知识图谱的一些信息，包括实体属性、实体关系等。文献[18]提出了一种简单有效的句向量计算方法，该方法使用 SIF(Smooth Inverse Frequency)权重计算词向量的加权平均，并将其作为初始句向量，利用 PCA 得到所有句向量的主成分，然后在其基础上减去某个句向量在第一主成分上的映射。该方法在多种 NLP 任务中的测试效果均优于复杂的神经网络方法，其同样可以用于事件向量的计算。

上述方法虽然可以用于计算开放域事件向量表征，但仍存在一些问题：首先，编码粒度过细，容易导致语义偏移；其次，这些方法并非直接对事件类型进行向量表征，而是以词向量为基础，通过各种组合运算间接地得到事件向量。由于局部语义之和不一定等于整体语义，因此这些方法对全局的事件相似或相关信息的捕捉能力较弱。

3 基于 Zipf's 共生矩阵分解的开放域事件向量计算

针对现有方法存在的问题，本文提出了基于 Zipf's 共生矩阵分解的开放域事件向量计算方法。该方法主要包含两部分：1）开放域事件标签提取；2）基于 Zipf's 共生矩阵分解的事件向量计算。

3.1 开放域事件标签提取

定义事件标签为某类事件的唯一标记。如果把事件句的原始字符序列当作事件标签，则会导致“维度灾难”。因此，需要从事件句中提取最能表达某种事件类型的关键信息。

ACE 将事件定义为由触发词和论元组成的结构，其中触发词是句子中最能表示事件发生的词。S_1 给出了一个事件描述，其触发词为“stole”，表明该事件为偷盗事件。ACE 定义的触发词多为动词，也有部分形容词或名词。论元包括时间、地点、参与者、数值等，事件参与者又可以分为事件发起者和承受者。S_1 中偷盗事件发生的时间为“yesterday”，偷盗的物品为“a wallet”，偷盗者为“Brown”，发生的地点为“ in the supermarket”。

S_1：Brown stole a wallet in the supermarket yesterday.

事件句中最能表达事件类型的元素是触发词，很多事件检测方法通过检测触发词来对事件类型进行判断。由于词汇存在一词多义现象，因此同一词汇可能触发不同的事件类型，例如，S_2 中的“beat”一词有两种常见的释义——“殴打”和“击败”。如果只考虑触发词，将无法判断 S_2 的真实事件类型。而如果同时考虑论元信息，例如已知“Anderson”和“Jane”是总统候选人，且事件发生在总统选举的过程中，那么“beat”在 S_2 中的含义更可能为“击败”。鉴于此，很多事件抽取方法在进行事件检测时，会同时考虑事件的论元信息。而在事件论元中，事件参与者往往包含更多的事件类型信息，因此本文从事件句中提取触发词和参与者来构成事件元组，并将其作为事件的类型标签。

S_2：Anderson beat Jane in the presidential election.

3.1.1 事件元组的定义

本文定义事件 e 的事件元组 t 为包含其触发词 p、事件发起者 a_s、承受者 a_o 的三元组，其结构为 $t=(a_s,p,a_o)$。

S_3：A drunk man broke the street lamp.

S_3 描述了一起“打碎东西”事件，对其进行依存分析的过程如图 1 所示。其中，“broke”是依存树的根，在事件句中作为触发词；“broke”的左右子树分别是事件的发起者“A drunk man”和承受者“ the street lamp”，在句子结构中分别对应主语和宾语。S3 对应的事件元组为（A drunk man，broke，the street lamp）。

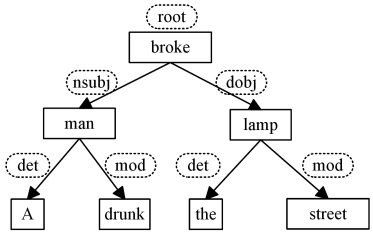

图 1 事件句 S_3 的依存分析图

Fig. 1 Dependency tree of S_3

3.1.2 事件元组的提取

本文主要依据依存分析的结果提取触发词和事件参与者。与 ACE 不同，本文只考虑动词性触发词，且不考虑系动词。在事件句的依存关系树中，动词性触发词通常是整个事件句的根（如 S_3 中的“bombed”），或者某个子事件句的根。从依存关系的角度分析，动词性触发词往往与主-谓关系或谓-宾关系同时出现，如 S_3 中的“nsubj”和“dobj”。因此，通过检测主谓宾关系可以找到动词性触发词。与此同时，与动词性触发词存在主谓宾关系的左右子树通常是事件的参与者，如 S_3 中的“A US warplane”和“a terrorist training camp”。

一些谓词，如“take” “get”等，往往与介词等组成固定搭配来表示特定的含义，如“take up” “get in”等。本文提取这些固定搭配作为触发词，利用下划线对各个词进行拼接，将其表示为形如“take_up”的形式。

根据配价理论，不同动词可能拥有不同的价数，常见的主要有 1，2，3 这 3 种价数。事件句 S_3 中“bombed”的价数为 2。事件句 S_4 中，触发词“cried”只有主语，没有宾语，价数为 1，对应的事件元组为（She，cried，UNK），其中“UNK”代表不存在或无法解析。事件句 S_5 中，触发词“gave”的价数为 3，对应的配价分别为“He”“me”和“a book”。从 S_5 中能够提取两个事件元组，分别为（He，gave，me）和（He，gave，a book），这两个元组拥有相同的触发词和事件发起者。

S_4：She cried.

S_5：He gave me a book.

一个句子可能包含多个触发词，因此可能触发多个事件。如果触发词之间是并列关系，并且主语相同，通常会省略主语。例如，事件句 S_6 中包含两个触发词，分别为“quit”和“started”，在依存树中，“He”与“started”之间没有显式的依存边，但由于“quit”和“started”是并列关系，推断其主语相同，对应的事件元组分别为（He，quit，his job）和（He，started，a business）。当触发词处于补语或状语中时（例如事件句 S_7），主语通常也是缺失的，此时需要依据语法关系对其进行补全。

S_6：He quit his job and started a business.

S_7：It rained for weeks，causing serious floods.

如果事件句以被动语态出现，则提取触发词的主语作为事件承受者，提取宾语作为事件的发起者，如提取 S8 的事件元组为（Russian warplanes，bombed，Some camps of Syrian rebels）。

S_8：Some camps of Syrian rebels were bombed by Russian warplanes.

3.1.3 事件元组的抽象

利用上述方法得到的事件元组过于实例化，无法概括事

件类型。例如，S_3 和 S_8 都描述了轰炸事件，但对应的元组差别较大，很难判断是否属于同一类型。本文利用 Zipf's 共生矩阵描述事件的上下文分布，由于 Zipf's 分布受采样次数的影响较大，如果元组在语料中的命中率较低，则将会影响事件向量的计算结果。因此，需要对元组进行进一步的抽象和加工。

为了减少触发词的种类，本文忽略时态差异，将触发词转化为其对应的词元，并将触发词中的所有单词转为小写，如“starts”“starting”和“started”都对应触发词“start”。

如果事件参与者是命名实体，则利用命名实体识别工具将其映射为对应的命名实体标签。例如，将“Trump”映射为“PERSON”(简写为“PER”)，将“New York”映射为“CITY”。为了与句子中的普通单词区别，将命名实体标签设置为大写。人称代词用标签“PER”代替，无法识别的命名实体用标签“NAME”代替。命名实体识别工具在检测不常见的实体时可能会出现误判，例如将地名、组织机构名等误判为人名。本文对可能的误判进行检测，并统一用标签“NAME”代替。

如果事件参与者是名词性短语，则去除所有的修饰语，只保留短语中的核心词，同时将核心词转化为词元。如果事件参与者是无法识别的代词，则用“UNK”代替。

以 S_3 和 S_8 为例，对事件参与者提取中心词，并对触发词词元化，得到相同的事件元组(warplane, bomb, camp)。可见，元组抽象能够在保持事件类型信息的前提下，提升元组在语料中的命中率，缓解样例稀疏问题。

3.1.4 事件元组剪枝

元组抽象能够在一定程度上缓解样例稀疏问题，但如果元组中包含生僻词，则同样会降低元组在语料中的命中率。以元组(al-Zarqawi, kill, PER)为例，由于“al-Zarqawi”为生僻词，该元组在语料中出现的次数较少。针对这种情况，本文将元组中的某些参与者用标签“UNK”代替，相当于删除了参与者在依存树中的对应子树，因此将该过程命名为元组剪枝。由于删除了生僻词，一些原先不相同的元组成为了相同元组，相当于增加了元组的命中率。

如何选择待删除的事件参与者，是元组剪枝需要考虑的首要问题。以(PER, make, debut)为例，“PER”在所有元组中出现的频率远高于“debut”，如果删除“debut”，则元组的语义将过于宽泛；而如果删除“PER”，则从(UNK, make, debut)依然可以推断该事件的类型为“亮相”。因此，元组剪枝应平衡以下两个目标。

(1)尽可能保持事件类型信息。

(2)增大元组命中率。

为了对元组剪枝进行全局优化，本文构建共生矩阵对事件参与者与触发词的共现情况进行统计。

设 $A=\{a_i \mid 1\leqslant i\leqslant |A|\}$，其中 a_i 为事件参与者的实体类型标签或核心词，$|A|$ 为两者的总类型数。设 $n(a_i)$ 为 a_i 在事件元组中的出现次数，定义 $A_{\text{trunc}}=\{a \mid a\in A, n(a)>\alpha_1\}$，$\alpha_1$ 为截断阈值。

设 $P=\{p_j \mid 1\leqslant j\leqslant |P|\}$，其中 p_j 为触发词，$|P|$ 为触发词的总类型数。设 $n(p_j)$ 为 p_j 在事件元组中出现的次数，定义 $P_{\text{trunc}}=\{p \mid p\in P, n(p)>\alpha_2\}$，$\alpha_2$ 为截断阈值。

设 $I(a)$ 和 $I(p)$ 分别为 a 和 p 在 A_{trunc} 和 P_{trunc} 中的索引。构建 $|A_{\text{trunc}}|\times|P_{\text{trunc}}|$ 维矩阵 $\boldsymbol{M}_{a-p}$，$\boldsymbol{M}_{a-p}$ 的 $I(a)$ 行对应事件发起者 a，$I(p)$ 列对应触发词 p。$\boldsymbol{M}_{a-p}[I(a), I(p)]$ 表示事件发起者为 a、触发词为 p 的元组出现的次数。

构建 $|P_{\text{trunc}}|\times|A_{\text{trunc}}|$ 维矩阵 $\boldsymbol{M}_{p-a}$，矩阵 $\boldsymbol{M}_{p-a}$ 的 $I(p)$ 行对应触发词 p，$I(a)$ 列对应事件承受者 a，$\boldsymbol{M}_{p-a}[I(p), I(a)]$ 表示触发词为 p、事件承受者为 a 的元组出现的次数。

对于任意元组 $t=(a_s, p, a_o)$，$a_s\in A_{\text{trunc}}$，$a_o\in A_{\text{trunc}}$，$p\in P_{\text{trunc}}$，设其在语料中的出现次数为 $n(t)$。

(1)若 $n(t)\geqslant\beta_1$(其中 β_1 为阈值)，则无须剪枝；否则进行下述步骤。

(2)若 $\boldsymbol{M}_{a-p}[I(a_s), I(p)]\geqslant\beta_2$，且同时满足 $\boldsymbol{M}_{p-a}[I(p), I(a_o)]\geqslant\beta_2$(其中 β_2 为阈值)，此时若 $\boldsymbol{M}_{a-p}[I(a_s), I(p)]\geqslant\boldsymbol{M}_{p-a}[I(p), I(a_o)]$，则删除 a_s；反之，删除 a_o。

(3)若 $\boldsymbol{M}_{a-p}[I(a_s), I(p)]<\beta_2$，且同时满足 $\boldsymbol{M}_{p-a}[I(p), I(a_o)]\geqslant\beta_2$，则删除 a_s。

(4)若 $\boldsymbol{M}_{a-p}[I(a_s), I(p)]\geqslant\beta_2$，且同时满足 $\boldsymbol{M}_{p-a}[I(p), I(a_o)]<\beta_2$，则删除 a_o。

(5)若 $\boldsymbol{M}_{a-p}[I(a_s), I(p)]<\beta_2$，且同时满足 $\boldsymbol{M}_{p-a}[I(p), I(a_o)]<\beta_2$，则删除 a_s 和 a_o，只保留触发词。

3.1.5 事件元组的消歧

若某个事件元组的触发词存在歧义，且参与者过于抽象，则该事件元组很可能存在歧义。例如，S_9 和 S_{10} 中均包含事件元组(PER, beat, PER)，根据语境可以判断它们属于不同的事件类型，如果不进行消歧，得到的语义将是两种语义的混合，且偏向较为常见的类型，对事件向量的性能将产生不利影响。

S_9: In 2017, he became Myanmar's first ever world champion in any mainstream sport, beating Vitaly Bigdash to win the ONE Championship middleweight title.

S_{10}: Thorne beats Sylvian and accuses him of stealing Linden from him.

人们对词语的真实含义产生疑惑时，会从上下文中寻找判断依据。同样地，对于有歧义的事件元组，本文考虑从上下文中寻找线索。由于上下文中既存在相关信息，也存在无关噪声，因此如何提取事件相关信息是事件消歧需要考虑的核心问题。

由于目标事件与上下文事件之间往往存在关联，因此可以通过上下文事件的类型推断目标事件的类型。例如，S_9 中出现了“赢得冠军”和“获得头衔”，据此可以推断 S_9 中“beat”为击败；而由于 S_{10} 中包含“指控”和“偷盗”，因此，S_{10} 中“beat”更可能为殴打。

从文档中抽取事件元组，得到事件元组的顺序列表 $T=\{t_i \mid 0<i\leqslant |T|\}$。将位于目标元组 t_i 前后大小为 2ω 的窗口内的事件元组集合 $T_c=\{t_{i-\omega}, \cdots, t_{i-1}, t_{i+1}, \cdots, t_{i+\omega}\}$ 作为其上下文，计算 T_c 的向量表征 $\boldsymbol{v}(T_c)$：

$$\boldsymbol{v}(T_c)=\frac{1}{2\omega}\sum_{j=1}^{2\omega}\boldsymbol{v}(t_j) \tag{1}$$

其中，$t_j\in T_c$；$\boldsymbol{v}(t_j)$ 为 t_j 的向量表征，计算公式为：

$$\boldsymbol{v}(t_j)=\sum_{w\in t_j}\lambda(w)\cdot\boldsymbol{v}(w) \tag{2}$$

其中，w 为 t_j 中的触发词或核心词；$\boldsymbol{v}(w)$ 为 w 的词向量；

$\lambda(w)$为其权重，本文采用 SIF[17] 全局权重（计算式如后文式(5)所示）。

假设事件元组 t'在语料中出现的总次数为 N。随机采样 $n(n<N)$次，分别计算 T_c 的向量表征，得到向量表征集合 $\boldsymbol{V}_{T_c}=\{\boldsymbol{v}_i \mid 1<i\leqslant n\}$。对 $\boldsymbol{v}_{T_c}$ 进行聚类，假设得到 m 个簇，则将 t'对应的事件类型细分为 m 个子类，每个子类对应一个中心点 $\boldsymbol{center}_j(1\leqslant j\leqslant m)$，以簇序号作为子类别标签。在统计事件共生矩阵时，对于 t'的每一次出现，计算其上下文事件表征 $\boldsymbol{v}(T_c)$，然后将上下文事件表征 $v(T_c)$与每个簇中心点做比较，计算 t'的子类别标签为：

$$l=\arg\max_j(\boldsymbol{center}_j,\boldsymbol{v}(T_c)) \tag{3}$$

元组在语料中出现的次数越多，意味着可采样次数也越多。本文只对语料中出现次数超过一定阈值的事件元组进行消歧，其原因为：

(1)元组出现越频繁，通常意味着语义越宽泛，越有可能出现歧义；

(2)本文方法需要保证一定的采样次数，而消歧过程将元组划分为多个子类别，会减少元组的可采样次数。

本文利用 K-Means 聚类算法对 $\boldsymbol{V}_{T_c}$ 进行聚类，将初始簇中心点的个数设置为 4，根据每个子类在语料中的命中次数确定是否保留该子类。

3.2 事件向量的计算

Harris 的分布假说认为：上下文分布相似的单词通常也拥有相近的语义。现有的词向量模型大多基于该假说。

上下文分布的表示方法主要有两种：共生矩阵和神经网络。对应地，词向量模型可以分为基于共生矩阵分解和基于神经网络模型两种。两种方法各有优劣：基于共生矩阵分解的方法统计和计算比较简单，但共生矩阵的维度较高，存储和降维比较困难；基于神经网络的方法对存储空间的消耗较少，但训练通常比较复杂且费时。

分布假说同样适用于事件向量表征。文献[3]提出了一种基于 Zipf's 共生矩阵分解的词向量计算模型，该模型大幅度减小了共生矩阵的维度，简化了共生矩阵的统计和矩阵项值的变换，缩短了计算时间，在多个任务中取得了接近于复杂神经网络模型的测试结果。与传统模型采用固定且较短的上下文窗口不同，该模型直接以语句为上下文，能够覆盖整个事件句，因此可以很方便地计算事件向量。

3.2.1 事件共生矩阵

共生矩阵(Co-occurrence Matrix)是对共现关系的一种矩阵描述，可以用于表示词汇、句子、文档等的上下文分布。本文采用共生矩阵描述事件的上下文分布。定义事件元组与上下文词的共生矩阵为 $\boldsymbol{M}_c$。$\boldsymbol{M}_c$ 的一行称为一个共生向量，对应某个事件元组 t；$\boldsymbol{M}_c$ 的一列代表某个上下文词 c，矩阵的一项代表某个上下文词 c 在某个事件元组 t 的上下文中出现的次数，记为 $n(t,c)$。元组 t 的上下文，即为 t 所在的整个事件句。

假设某个语料中只包含 S_3 和 S_4 两个事件句，对应的事件元组分别为(warplane，bomb，camp)和(PER，cry，UNK)，记为 t_1 和 t_2。将上下文词汇变为小写，按照出现次数逆序排列得到上下文词汇表 $C=\{$a，us，warplane，bombed，terrorist，training，camp，she，cried$\}$(对应表 1 中的 c_1 到 c_9)，则共生矩阵样例如表 1 所列。

表 1 共生矩阵样例

Table 1 Example of co-occurrence matrix

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
t_1	2	1	1	1	1	1	1	0	0
t_2	0	0	0	0	0	0	0	1	1

对大规模语料进行统计，从而得到高维稀疏的共生矩阵。由于词频分布遵循 Zipf's 定律，原始统计值之间可能相差多个数量级，一些高频词将拥有异常的高权重，如“a”“the”“of”等高频词几乎存在于所有上下文中，语义区分性较差，因此需要对矩阵值进行变换。

3.2.2 矩阵变换及分解

在词向量研究领域，研究人员提出了一些方法对矩阵值进行变换。Bullinaria 等[19]对几种变换方法分别进行了比较，认为采用 PPMI 变换可以压缩原始项值的差异性，得到的词向量性能最好；同时对上下文窗口大小和共生矩阵维度进行了研究，认为较小的上下文窗口和较高的矩阵维度能够提高词向量的性能。

然而，本文经过实验得到了截然不同的发现。首先，保持矩阵项的 Zipf's 特征对向量表征是有用的；其次，增大上下文窗口虽然会带来噪音，但也会带来更丰富的语境信息；最后，只需要少量的常见词就可以有效地表示上下文分布。本文提出了一种新的矩阵变换方法，该方法保持了矩阵项的 Zipf's 特征，同时降低了高频词的权重，对较大的上下文窗口有更好的抗噪性，同时大幅度缩小了共生矩阵的维度。

本文首先对 Zipf's 定律进行介绍。Zipf's 定律是语言学家 Zipf's 提出的一种词频分布定律，它揭示了人类语言所遵循的“Least Effort”原则，即单词表中只有少量单词会频繁使用，而大部分单词的使用次数很少。例如，对英文 Wikipedia 2018 语料进行统计，其中总的词汇量大约为 20 亿，词汇类型超过 1 000 万。对词频按照逆序排列，其中前 6 000 个词(词形)的词频总数约占总词汇量的 80%以上。

Zipf's 定律说明，人类只需要少量的词汇就可以表述非常丰富的信息。受此启发，本文认为少量的词汇同样可以有效地表示事件的上下文分布。为了保持矩阵项的 Zipf's 特征，并消除高频词的高权重，本文引入了一种新的矩阵变换方法：

$$n'(t,c)=\left(\frac{n(t,c)+\varepsilon}{n(c)}\right)^r \tag{4}$$

其中，$n(t,c)$是 c 出现在事件元组 t 的上下文中的次数，$n(c)$是 c 在整个语料中出现的次数。$n(c)$，$n'(t,c)$和 $n(t,c)$三者均表现出了 Zipf's 分布特征。由于 $n(t,c)$就越大，$n(c)$越大，且 $n(c)$的变化更快，因此高频词的 $n'(t,c)$值通常较小。$n(c)$往往远大于 $n(t,c)$，导致 $n'(t,c)$的变化比较陡峭。本文采用平滑指数 $r(0<r<1)$对其进行平滑。为了避免矩阵中出现 0 值，对每个原始统计值 $n(t,c)$增加一个正值 ε。

不同词汇对上下文分布的贡献程度不同。为了进一步优化 $n'(t,c)$的取值，本文为每一个上下文词汇赋予全局权重。SIF 是一种简单有效的全局加权方法，其计算公式为：

$$\lambda(c)=\frac{a}{a+f(c)} \tag{5}$$

其中，a 为常数，$f(c)$是 c 出现在语料中的频率，$\lambda(c)$随着

$f(c)$的增大而减小。利用$\lambda(c)$对$n'(t,c)$进行加权，得到$n_\lambda'(t,c)$。

$$n_\lambda'(t,c)=\lambda(c)\times n'(t,c) \tag{6}$$

由于$n_\lambda'(t,c)$表现出了 Zipf's 分布特征，本文将变换后的矩阵称为 Zipf's 共生矩阵。相比传统共生矩阵，Zipf's 共生矩阵存在一些优良的特性。

(1)抗噪性强适用于较大的上下文窗口。由于幂率分布(Zipf's 分布的本质是一种幂率分布)存在"富者益富"现象，对于 Zipf's 共生矩阵，较大的上下文窗口虽然会引入噪音信息，但相关分量的增长更快。

(2)更小的矩阵维度。相比传统的共生矩阵，Zipf's 共生矩阵的维度大幅度缩减(对 3000 维 Zipf's 共生矩阵进行分解、变换后得到的词向量在多个任务中取得了与复杂神经网络词向量相近的性能)。降低矩阵维度，能够减小存储和计算开销，同时缩短统计时间。

(3)直接使用整个事件句作为上下文，在提升向量性能的同时也省略了传统的滑动窗口，缩短了统计时间。

对于矩阵分解，本文首先采用 PCA 对共生矩阵进行降维，得到初始向量。由于 PCA 无法提取特征中的非线性成分，本文构建自编码器对初始向量进行非线性变换，得到最终的事件向量。这一部分内容沿用文献[3]中的方法。

4 实验分析

4.1 事件元组提取

本文从英文大规模开放语料 Wikipedia 2018 dump[1)]中提取事件元组及对应的事件句。采用 Stanford NLP 工具进行分句及依存分析；在依存分析的基础上，利用本文方法进行事件元组提取，得到初始的事件元组，总量约为 1.37 亿个，种类约为 3885 万种，对应事件句约为 7656 万条。原始的事件参与者列表 A 的大小约为 1.7×106，触发词列表 P 的大小约为 1×104。设置参数 α_1，α_2，β_1 和 β_2 分别为 200，300，600，300，得到$|A_{\text{trunc}}|$的大小约为 2.6×104，$|P_{\text{trunc}}|$的大小约为 7600，对初始元组进行抽象和剪枝后，得到事件元组总量约为 1.07 亿个，类型约为 7.2 万种，对应事件句为 6553 万条。本文针对出现次数最多的前 7 万种事件元组计算事件向量。

4.2 事件向量生成

对 Wikipedia 语料进行词频统计。去除含有数字或符号的词汇，将所有的词汇词元化并转化为小写。将得到的词元按照词频大小逆序排列，选择前 3000 个词元作为上下文特征。计算特征的 SIF 权重，生成词汇到上下文特征的映射词典。

初始化共生矩阵，矩阵维度为 70000×3000；然后对共生矩阵进行统计。由于取消了滑动窗口，同时限制了元组的采样次数，因此共生矩阵的统计效率得到了提高，总的统计时长约为 26 min。

对共生矩阵进行变换，然后利用 PCA 对共生矩阵进行降维，得到初始事件向量，总时长约为 40 s。

构建自编码器对初始事件向量进行非线性变换，自编码器主要包含编码器和解码器两部分。其中，编码器对事件向量 $\boldsymbol{v}_{\text{true}}$ 进行编码；解码器对编码结果进行解码，重构事件向量 $\boldsymbol{v}_{\text{pred}}$。隐层采用 ReLU 激活函数，输出层采用 Tanh 激活函数。利用 AdaMax 优化器最小化 $\boldsymbol{v}_{\text{true}}$ 与 $\boldsymbol{v}_{\text{pred}}$ 的余弦损失，学习率 lr 设置为 0.003，总时长约为 18 min。

$$\cos\theta=\frac{\sum_1^n(\boldsymbol{v}_{\text{true}}^{(i)}\times\boldsymbol{v}_{\text{pred}}^{(i)})}{\sqrt{\sum_1^n(\boldsymbol{v}_{\text{true}}^{(i)})^2}\times\sqrt{\sum_1^n(\boldsymbol{v}_{\text{pred}}^{(i)})^2}}$$
$$L=-\frac{1}{m}\sum_{j=1}^m\cos_j\theta \tag{7}$$

4.3 最近邻检测

事件表征将事件的语义相似性映射为向量的空间相邻性，通过最近邻检测能够比较直观地显示事件表征的效果。表 2－表 4 列出了与目标事件元组最相邻的 Top-5 个事件元组(表中对一些命名实体标签进行了缩写)。检测结果表明，本文方法能够有效表征事件元组之间的语义相似性。此外，由于事件句中往往包含多个相关的事件元组，而这些元组共享上下文，因此最近邻中不仅包含语义相似的元组，还包含语义相关的元组。实验结果证明，本文方法能够同时获取事件的关联信息。

表 2 最近邻分析结果(1)

Table 2 Results of nearest neighbors (1)

PER,die_of,cancer	PER,serve_as,chairman	PER,sentence_to,year	PER,know_for,role	PER,travel_to,CITY
PER,die_from,cancer	PER,become,chairman	UNK,sentence_to,year	UNK,appear_in,drama	PER,travel_to,LOC
UNK,die_of,cancer	UNK,serve_as,chairman	PER,serve,sentence	PER,know_for,portrayal	PER,travel_with,PER
UNK,die_from,cancer	UNK,appoint_as,chairman	UNK,sentence_to,prison	PER,star_in,movie	PER,come_to,CITY
UNK,battle,cancer	PER,serve_as,secretary	court,sentence,UNK	PER,co－star_in,film	PER,visit,CITY
mother,die_of,UNK	PER,appoint_to,board	UNK,sentence_to,term	PER,star_in,drama	PER,make,acquaintance

表 3 最近邻分析结果(2)

Table 3 Results of nearest neighbors (2)

UNK,play,football	election,hold_on,DATE	UNK,write,book	PER,receive,bachelor	PER,leave,ORG
UNK,play,basketball	UNK,elect_for,term	UNK,author,book	PER,earn,bachelor	PER,leave,position
UNK,play,baseball	PER,run_for,reelection	UNK,publish,book	UNK,receive,bachelor	PER,leave,post
UNK,play,hockey	election,coincide_with,UNK	PER,publish,book	PER,have,bachelor	UNK,join,ORG
UNK,play_for,team	PER,elect_to,House	PER,write,book	PER,complete,bachelor	PER,serve_as,manager
UNK,play,soccer	PER,elect_to,term	UNK,write,series	PER,graduate_with,TITLE	PER,become,vice-president

1) https://dumps.wikimedia.org/

表 4 最近邻分析结果(3)

Table 4 Results of nearest neighbors (3)

PER,bury_in,UNK	PER,take,job	UNK,establish,ORG	team,win,championship	UNK,reduce,cost
PER,bury_at,UNK	PER,hold,job	UNK,create,ORG	team,win,tournament	UNK,lower,cost;
PER,bury_on,DATE	PER,get,job	UNK,set_up,ORG	UNK,win,conference	UNK,reduce,need
UNK,bury_in,graveyard	PER,find,job	UNK,propose,creation	UNK,earn,championship	UNK,increase,efficiency
PER,bury_in,cemetery	PER,find,work	PER,establish,ORG	NAME,win,championship	UNK,cut,cost
PER,bury_with,UNK	PER,land,job	ORG,establish,UNK	UNK,win,championship	UNK,improve,efficiency

利用最近邻检测对事件元组消歧的结果进行分析，如表 5 所列。以事件元组(PER,beat,PER)为例，在消歧前，元组的语义偏向于“击败”，最近邻多为“获胜”“输掉”“赢得”等。对元组进行消歧后，得到 3 个子类别。其中，前两个子类别是对“击败”这一语义的细化，而第三个子类别的语义应为“殴打”，这是由于其最近邻中出现了“殴打(beat_up)”“攻击”“复仇”等事件。这一结果证明了本文消歧方法的有效性。

表 5 事件元组消歧结果

Table 5 Results of event tuple disambiguation

消歧前	消歧后		
PER,beat,PER	PER,beat,PER,1	PER,beat,PER,2	PER,beat,PER,3
PER,beat_by,PER	UNK,reach,final	PER,finish_as,runner-up	PER,beat_up,UNK
PER,lose,final	PER,finish_as,runner-up	UNK,beat_by,PER	PER,get,revenge
PER,lose,match	UNK,win,first	UNK,complete,double	PER,rape,PER
UNK,win,second	PER,reach,quarter-final	UNK,beat,rival	PER,attack,PER
PER,reach,quarter-final	UNK,complete,double	UNK,win,Cup	PER,kidnap,PER

4.4 事件检测

事件检测是事件抽取的一项基本任务。由于现有事件检测数据集的规模较小，本文选择 FrameNet[20] 数据作为事件检测数据。FrameNet 定义的框架结构与事件结构很相似，每种框架(Frame)包含若干 LU(Lexical Unit)和 FE(Frame Element)。LU 是语句中最能表示框架类型的词，相当于事件中的触发词。FE 表示框架中的各种语义角色，相当于事件中的论元。由于框架与事件之间具有相似性，一些事件检测方法利用 FrameNet 扩充事件检测数据集。

每个 LU 对应若干例句，这些例句可以作为事件检测的标记样例。本文随机选择 200 种框架和 43 564 个标记样例，将这些标记样例随机划分为训练集、测试集和验证集，对应样例数量分别为 32 384，6 367 和 4 813。本文选择了几种有代表性的方法，用于计算开放域事件向量。

(1)计算句子中所有单词向量的非加权平均，简记为“Unweighted”。

(2)使用 SIF 句向量方法生成事件向量，简记为“SIF”。

(3)使用 Bert[1)] 模型[21] 的句向量计算功能直接生成事件向量，简记为“Bert”。

(4)利用文献[22]提出的依存树递归自编码器生成事件向量，简记为“DT-RAE”。

(5)将触发词词向量作为事件向量，简记为“Trigger”。

(6)将触发词与论元核心词词向量的非加权平均作为事件向量，简记为“Trigger+Argument”。

(7)利用本文方法从句子中提取事件元组，查询元组词典得到事件向量。由于元组消歧需要提取上下文事件，而事件检测大多面向单个句子，因此本文事件检测任务使用的是未消歧的事件向量。

本文利用 Zipf's 共生矩阵分解法生成词向量，语料为英文 Wikipedia 2018 语料，利用全连接神经网络+Softmax 多分类器构建事件检测模型。网络分为输入层、隐藏层和输出层 3 层，隐藏层采用 ELU 激活函数，输出层采用 Softmax 激活函数。采用 Nadam 优化器优化真实类别与预测类别之间的交叉熵损失，学习率 lr 设置为 0.001，采用“Early Stopping”策略防止过拟合，评价指标为准确率的加权平均 P、召回率加权平均 R，以及 F_1 值。

表 6 比较了几种向量生成方法。其中，前 4 种方法在很多 NLP 任务中有较好的表现，但由于编码粒度过细，出现了语义偏移现象，在事件检测中的效果并不理想。单纯考虑触发词能够取得较好的性能，体现了触发词在事件检测中的关键作用。同时，简单地计算触发词和论元核心词的向量平均并不能提高事件检测效果，证明整体的语义不一定等于各部分语义之和。本文方法将事件元组作为一个整体来计算事件向量，避免了编码粒度过细的问题，能够从全局角度表征事件相似性和相关性，因此在事件检测任务中表现出了较好的效果。

表 6 事件检测结果的比较

Table 6 Comparison of different event embeddings on task of event identification

(单位：%)

Method	Test			Dev		
	P	R	F_1	P	R	F_1
Unweighted	64.4	63.2	63.8	66.7	65.1	65.9
SIF	66.2	65.7	65.9	68.7	67.7	68.2
DT-RAE	66.0	65.4	65.7	65.9	65.0	65.4
Bert	67.6	64.9	66.2	69.4	65.9	67.6
Our Method	84.1	83.3	83.7	84.7	83.7	84.2
Trigger	79.4	78.6	79.0	80.0	78.8	79.4
Trigger+Argument	74.1	72.3	73.2	74.9	73.1	74.0

结束语 针对大规模开放领域事件向量表征困难的问题，本文提出了一种基于 Zipf's 共生矩阵分解的事件向量计算方法。该方法首先从大规模语料中提取事件元组，然后对

1) https://dumps.wikimedia.org/arch/bert

元组进行抽象、剪枝和消歧；使用 Zipf's 共生矩阵表示事件元组的上下文分布，利用 PCA 对矩阵进行降维，并利用自编码器对事件向量进行非线性变换。实验部分采用最近邻检测和事件检测对事件向量的性能进行了评估。实验结果表明，本文方法从整体上对事件元组进行向量表征，避免了语义偏移，能够从全局角度表征事件之间的相似性和相关性。

现有的 NLP 工具在依存分析和实体类型检测方面的性能不够理想，影响了事件元组提取以及后续事件向量的计算。未来将对这一问题进行深入探索。

参 考 文 献

[1] CHEN Y, LIU S, ZHANG X, et al. Automatically labeled data generation for large scale event extraction[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver, Canada: ACL, 2017: 409-419.

[2] HARRIS Z S. Distributional structure[J]. Word, 1954, 10(2/3): 146-162.

[3] GAO L, ZHOU G, LUO J, et al. Word Embedding With Zipf's Context[J]. IEEE Access, 2019, 7: 168934-168943.

[4] MIKOLOV T, CHEN K, CORRADO G, et al. Linguistic regularities in continuous space word representations[C]// Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Seattle, USA: ACL, 2013: 746-751.

[5] PENNINGTON J, SOCHER R, MANNING C. Glove: Global vectors for word representation[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: ACL, 2014: 1532-1543.

[6] BOJANOWSKI P, GRAVE E, JOULIN A, et al. Enriching word vectors with subword information[J]. Transactions of the Association for Computational Linguistics. Vancouver, Canada: ACL, 2017: 135-146.

[7] NGUYEN T H, GRISHMAN R. Event detection and domain adaptation with convolutional neural networks[C]// Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Beijing, China: ACL, 2015: 365-371.

[8] CHEN Y, XU L, LIU K, et al. Event extraction via dynamic multi-pooling convolutional neural networks[C]// Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Beijing, China: ACL, 2015: 167-176.

[9] NGUYEN T H, GRISHMAN R. Graph convolutional networks with argument-aware pooling for event detection[C]// Thirty-Second AAAI Conference on Artificial Intelligence. Louisiana, USA: AAAI, 2018.

[10] SHA L, QIAN F, CHANG B, et al. Jointly Extracting Event Triggers and Arguments by Dependency-Bridge RNN and Tensor-Based Argument Interaction[C]// Thirty-Second AAAI Conference on Artificial Intelligence. Louisiana, USA: AAAI, 2018.

[11] ORR J W, TADEPALLI P, FERN X. Event Detection with Neural Networks: A Rigorous Empirical Evaluation[C]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: ACL, 2018: 999-1004.

[12] ZHANG J, QIN Y, ZHANG Y, et al. Extracting entities and events as a single task using a transition-based neural model[C]// Proceedings of the 28th International Joint Conference on Artificial Intelligence. Hawaii, USA: AAAI, 2019: 5422-5428.

[13] NGUYEN T M, NGUYEN T H. One for all: Neural joint modeling of entities and events[C]// Proceedings of the AAAI Conference on Artificial Intelligence. Hawaii, USA: AAAI, 2019: 6851-6858.

[14] LIU J, CHEN Y, LIU K. Exploiting the Ground-Truth: An Adversarial Imitation Based Knowledge Distillation Approach for Event Detection[C]// Proceedings of the AAAI Conference on Artificial Intelligence. Hawaii, USA: AAAI, 2019, 33: 6754-6761.

[15] HUANG L, CASSIDY T, FENG X, et al. Liberal event extraction and event schema induction[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: ACL, 2016: 258-268.

[16] DING X, ZHANG Y, LIU T, et al. Deep learning for event-driven stock prediction[C]// Twenty-fourth international joint conference on artificial intelligence. Menlo Park, CA: AAAI, 2015.

[17] DING X, ZHANG Y, LIU T, et al. Knowledge-driven event embedding for stock prediction[C]// 26th International Conference on Computational Linguistics: Technical Papers. Osaka, Japan: COLING, 2016: 2133-2142.

[18] ARORA S, LIANG Y, MA T. A simple but tough-to-beat baseline for sentence embeddings[C]// Proc. ICLR. Toulon, France, 2017.

[19] BULLINARIA J A, LEVY J P. Extracting semantic representations from word co-occurrence statistics: A computational study[J]. Behavior Research Methods, 2007, 39(3): 510-526.

[20] BAKER C F, FILLMORE C J, LOWE J B. The berkeley framenet project[C]// Proceedings of the 17th International Conference on Computational Linguistics. Association for Computational Linguistics. Quebec, Canada: ACL, 1998: 86-90.

[21] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[C]// Proc. NAACL. 2019: 4171-4186.

[22] RANA D S, MISHRA P K. Paraphrase Detection using Dependency Tree Recursive Autoencoder[C]// 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS). IEEE, 2019: 678-683.

GAO Li-zheng, born in 1990, doctoral student. His research interests include information extraction and data mining.

ZHOU Gang, born in 1974, Ph. D, research fellow, professor. His research interests include big data and data mining.