中国科学: 信息科学 2020 年 第 50 卷 第 7 期: 1033–1054
SCIENTIA SINICA Informationis
领域大数据知识图谱专题 • 论文

基于篇章级事件表示的文本相关度计算方法

刘铭[1,2], 郑子豪[1], 秦兵[1,2*], 刘一仝[3], 李阳[3]

1. 哈尔滨工业大学计算机科学与技术学院, 哈尔滨 150001
2. 鹏城实验室, 深圳 518066
3. 腾讯科技 (北京) 有限公司, 北京 100193
* 通信作者. E-mail: qinb@ir.hit.edu.cn

收稿日期: 2019–12–21; 修回日期: 2020–03–29; 接受日期: 2020–05–13; 网络出版日期: 2020–07–13

科技创新 2030 —— "新一代人工智能" 重大项目 (批准号: 2018AAA0101901)、国家重点研发计划项目 (批准号: 2018YFB100-5103)、国家自然科学基金重点项目 (批准号: 61632011)、国家自然科学基金面上项目 (批准号: 61772156, 61976073) 和黑龙江省面上项目 (批准号: F2018013) 资助

摘要 随着网络信息的剧增, 信息流服务备受用户关注. 在信息流服务中, 如何衡量文本之间的相关度进而从多来源的信息渠道中过滤掉冗余信息提升推荐满意度成为至关重要的环节. 当前主流的文本相关度计算方法均是将文本表示为向量, 进而通过衡量向量之间的相似度来度量文本间的相关度. 然而, 信息流中的文本多为新闻文本, 这些文本的核心是其描述的事件, 基于此需要从事件的角度挖掘文本的核心特征进而利用其计算文本间的相关度. 当前针对事件的研究大多数着眼于句子级别. 事实上, 在计算文本相关度时, 需要从篇章级别把握文章的内容. 故此, 篇章级的事件分析更有影响力. 为此, 本文在句子级事件抽取的基础上, 提出了一种篇章级的事件表示方法, 其利用句子级事件的抽取结果构建篇章事件连通图, 并选取图中重要的节点作为篇章级事件的代表, 之后利用篇章级的事件表示结果来度量文本之间的相关度. 实验显示, 本文提出的文本相关度计算方法要远好于传统的文本相关度计算方法.

关键词 篇章事件连通图, 篇章级事件相关度, 文本排序, 关键子句筛选, 子句连通图

1 引言

随着网络信息的丰富, 信息流服务逐渐成为当前各大网络公司关注的焦点. 信息流服务的特点在于将多来源的信息整合在一起, 而多来源的信息中必然包含大量的冗余文本. 基于此, 需要一种方法能够有效度量文本之间的相关度, 进而过滤掉冗余文本. 当前信息流服务中的文本多为新闻文本, 而新闻文本的特点在于围绕着新闻事件展开, 其核心是对一个新闻事件的发展脉络进行介绍. 因此, 在

引用格式: 刘铭, 郑子豪, 秦兵, 等. 基于篇章级事件表示的文本相关度计算方法. 中国科学: 信息科学, 2020, 50: 1033–1054, doi: 10.1360/SSI-2019-0272
Liu M, Zheng Z H, Qin B, et al. Text correlation calculation based on passage-level event representation (in Chinese). Sci Sin Inform, 2020, 50: 1033–1054, doi: 10.1360/SSI-2019-0272

度量新闻文本间的相关度时, 必须要从事件的角度入手, 从篇章级的角度入手, 通过度量两篇新闻文本描述的事件的相关性进而获得文本间的相关度.

事实上, 篇章级事件是最能代表篇章核心内容的事件. 篇章级事件不一定是出现频率最高的事件, 但篇章中的所有事件, 一定都是围绕此核心事件展开的, 即篇章中的所有事件都一定与核心事件有关联. 一篇文章往往描述了多个事件, 简单叠加句子级事件抽取的结果无法获取篇章的核心事件, 这显示了研究篇章级事件抽取和表示的重要性. 好的篇章级事件抽取模块可以更好地对篇章进行建模, 能够从篇章的角度衡量两篇新闻文本是否描述同一事件, 从而更好地服务于文本相关度计算, 进一步提高用户的满意度.

传统的事件抽取方案一般聚焦于句子级别, 所采用的方法大致可分为 3 类: (1) 基于句法分析和模板匹配的事件抽取; (2) 融合句法信息及语义信息, 采用传统分类算法进行触发词以及事件元素的识别; (3) 融合句法信息、语义信息及位置信息的基于深度学习的事件触发词及事件元素识别.

第 1 类方法的通用思想是对句子进行句法分析, 将提取出的状动、动补等结构作为事件触发词, 将提取出的主语、宾语等当作事件元素 [1]. 或者更加复杂, 考虑句法树结构, 融合依存句法分析和短语结构句法分析, 按照预先设定的提取规则从句法树中进行事件触发词和事件元素的识别. 取得较好效果的基于规则的事件提取方法有通过启发式方法来构建事件模板, 以及通过种子模板在语料库中增量式的构建事件模板等方式 [2~7]. 虽然这些论文的目的在于生成多样化的事件模板用以覆盖各种类型的事件, 但是, 为了避免生成的模板过于发散, 均需要构建一定数量的种子模板作为启动项, 且生成的模板和种子模板的分布不应有太大差异.

第 2 类方法融合上下文、词法、词典特征, 利用预先标注的训练数据训练分类器以对触发词和事件元素进行多次二分类, 从而确定触发词和事件元素的类别. 但是这种方式将句法树扁平化处理, 忽略了树结构信息. 相对的, 有些工作则将句法分析结果转化为结构特征, 采用卷积树核函数对特征进行映射, 文献 [8] 即利用 SVM 分类器对触发词和事件元素进行多次二分类, 从而确定触发词和事件元素的类别. 之后有大量的研究者考虑融合命名实体、共现词, 以及短语结构等信息丰富事件特征 [9~12], 更有研究者利用跨语言、跨事件信息建模事件特征 [13~15], 并从全局和局部两个方向编码事件特征 [16].

第 3 类方法采用深度学习方法, 将词语的词向量、所在句中位置的编码向量, 以及句法角色编码向量等长向量拼接成特征矩阵, 之后采用 CNN 或 RNN 进行触发词和事件元素的识别和分类 [17~19]. 近年来, 随着深度学习技术的快速发展, 一些新的方法, 如: 注意力机制、预训练语言模型、图卷积网络等也被用于解决不同的事件挖掘问题. 文献 [20~22] 即分别利用多语言注意力机制来融合多种语言信息, 以及利用预训练语言模型 Bert 和图模型更好地建模文本表示, 获取事件元素之间以及事件元素和事件触发词之间的远程联系. 基于深度学习的事件挖掘或事件表示方法通过训练大规模的参数能够很好地建模事件的分布情况, 但是其缺点在于需要大规模的训练数据, 而现实中很难获取充足的已标注训练语料, 尤其是面向事件挖掘或事件表示这类复杂的任务.

通过上面的介绍可见, 早期的事件挖掘或事件表示方法多为无监督类型, 其依靠词法、句法等结构信息挖掘事件触发词和事件元素, 但是受限于词法、句法算法的性能, 无监督的事件挖掘或事件表示方法难以达到较高的准确率. 在此背景下, 有监督的事件挖掘或事件表示方法成为主流. 借助于事先标注的训练数据可以获得高质量的事件挖掘结果. 深度学习算法的出现进一步提升了事件挖掘和事件表示的准确度, 但是受限于深度学习算法需要训练大规模的参数这一特性, 其对标注语料的需求量进一步提升. 事实上, 在很多任务或者领域中很难获取大规模的标注语料. 以本文所针对的篇章级事件表示和基于此的文本相关度计算为例, 不同的用户对于一篇文章的核心事件可能有不同的理解, 这种情况会造成标注人员得到的标注结果很难保证高度的一致性, 从而降低了语料的质量, 进而降低了

依据语料所训练出的算法的性能. 基于此, 近些年研究者又将目光转向如何以无监督或半监督的方式建模事件的内部结构, 例如 Peng 等 [23] 将事件抽取任务转化为事件提及与事件本体类型的相似度计算问题, 利用大规模非事件类的标注数据建模事件抽取和事件共指之间联系, 利用最小化的监督信号缓解事件类标注语料规模过小的问题. 然而上述方法的缺点是缺乏建模事件间联系的方案, 其将一篇文章中描述的多个事件看作是相互独立的.

句子级事件挖掘方法提高了计算机对句子的理解程度, 将一句话中的关键信息以事件的形式提取出来, 对许多研究工作都具有十分重要的意义. 然而, 在很多情况下, 句子级事件过于细粒度, 无法满足实际需求, 比如新闻推荐, 一篇新闻会包含多个事件, 然而, 在过滤相似的新闻时应聚焦于新闻的核心事件. 此时, 就无法仅依靠句子级事件的提取结果了, 因为如果把所有的句子级事件的集合当作篇章级事件的代表会引入大量噪声. 再比如篇章级的逻辑推理, 即, 由某一事件引发了另一事件的产生, 若以句子级事件为操作对象会过于微观, 推理路径可能会非常长, 这样本来因果十分强的两个事件很可能因为较长的推理路径而减弱了因果相关度, 并且大量的可作为背景的新闻资料也是以篇章规模来描述事件. 如果能将句子级事件的推理过程省略, 直接计算篇章级事件的相关度, 将大大提高因果事件的推理准确度.

传统的事件挖掘或事件表示方法几乎都是面向句子级的, 其视多个句子描述的事件之间是无关的, 这使得句子级事件的提取方法很难直接应用到篇章级事件的相关应用中. 句子级事件提取一般是将句子的词义、语义、句法结构转化为向量, 再使用分类器对事件触发词和事件元素进行分类. 但是从篇章级别看, 一篇文章中包含的信息过多, 如果仅仅是将提取出的句子级事件叠加转换成篇章级事件的代表势必会丢失很多关键信息. 反之, 如若表示为矩阵形式, 则由于多个句子级事件很难共享大量类似的事件元素而使整个矩阵过于稀疏. 基于此, 本文提出了基于图的篇章级事件表示和文本相关度计算方法. 对于篇章级的事件表示, 由于一篇文章中会描述多个事件, 故而本文引入图结构, 通过图结构不仅能够刻画事件内部触发词和事件元素之间的联系还能够揭示跨事件之间的联系.

无论是句子级的事件挖掘问题还是本文提出的篇章级的事件表示任务都是意图对文本进行结构化表示. 例如在句子级的事件抽取任务中, 事件以事件触发词为核心, 事件元素依附于触发词而形成类似于表格的结构化形式. 而在本文的篇章级事件表示中则是通过图结构来刻画事件内部乃至事件之间的信息. 另外一种表示文本的方式则是可以利用无结构的压缩的文本片段, 即文本摘要. 文本摘要的目的是将文本的核心话题以精简的话语描述出来, 从而过滤掉文本中无关的信息. 基于此, 也可以利用文摘后的文本片段作为基准来计算文本相关度. 但是由后续的实验结果可以看出, 相较于本文提出的文本相关度计算方法, 这种基于文摘的方法得到的计算结果的准确率较低. 其原因在于, 文本摘要能够提取出覆盖核心事件的文本片段, 但是其缺点在于文本中除核心事件外, 其他事件也能够帮助丰富核心事件的内容, 包括帮助陈述事件的发展脉络和事件产生的原因, 这些信息在摘要中均被省略了, 从而造成摘要无法充分揭示文本的核心事件, 同时无结构的文本摘要中也会包含一些无用信息, 包括停用词、形容词等, 其对计算文本间的相关度没有任何帮助.

2 方法描述

本文的目的在于利用篇章级的事件表示结果来计算文本间的相关度, 其篇章级的事件表示以句子级事件抽取结果为基础, 通过图结构将句子内部事件触发词和事件元素之间的联系和跨句子事件间的联系融合到一起. 具体而言, 篇章级的事件表示以句子级事件为基础, 将句子级事件的触发词和事件元素作为节点, 首先构造句子级事件多边形, 再根据节点共现将事件多边形连接起来构造篇章级的事

件连通图. 此图能够从整体上反映出文档中包含的事件的分布情况. 之后, 使用 TextRank 算法确定篇章事件连通图中节点的权重, 根据权重提取出相对于文档来说最重要的若干节点. 将从篇章事件连通图中提取出的权重较高的节点作为篇章级事件的代表, 通过计算两篇文档对应的节点集合的重合度来确定两篇文档的篇章级事件相关度. 以下章节详细介绍了本文提出的方法所涉及的多个环节.

2.1 构建子句连通图

与短句不同, 一篇文章包含的信息过于冗杂. 比如一篇新闻, 除了对核心事件进行报道外, 还会对相关事件、人物甚至历史进行简略介绍, 以期让读者对新闻报道的背景有更加详尽的了解. 这对提取文章的核心事件引入了不小的噪声. 为使核心事件的提取更加准确, 首先要进行核心信息的提取, 将冗余信息过滤掉. 本文采用基于 EM (expectation maximization) 思想的 TextRank 算法对核心信息进行提取 [24]. 主要方案是: 构建以子句为节点的连通图, 将对文本信息有充分覆盖能力的子句提取出来. 首先将文档按照逗号、句号、感叹号、问号、回车换行符等符号分隔成若干子句, 对每个子句分词后去停用词. 将每个子句作为连通图中的一个节点, 若两个子句包含一个以上重合词, 则将两个节点连接起来, 构成一个连通图. 边权初始化公式如下式所示:

$$\mathrm{Score}\,(V_1, V_2) = \frac{|\mathrm{Intersection}\,(V_1, V_2)|}{|\mathrm{Union}\,(V_1, V_2)|}, \tag{1}$$

其中, V_1 和 V_2 分别代表两句话的节点, $\mathrm{Score}(V_1, V_2)$ 代表连接 V_1 和 V_2 间的边的初始边权大小, $|\mathrm{Intersection}(V_1, V_2)|$ 代表两个子句的词集合的交集中的词语个数, $|\mathrm{Union}(V_1, V_2)|$ 代表两个子句的词集合的并集中的词语个数.

假设, 一篇文章包含如下 4 句话:

- “特朗普携夫人访问英国.”
- “特朗普与首相共同接受记者采访.”
- “女王在白金汉宫举办欢迎晚宴.”
- “首相在晚宴上致词感谢特朗普夫妇的来访.”

生成的子句连通图如图 1 所示, 图中的节点是上述 4 句话分词后的结果. 边权的初始化方法是: 被连接的两个节点句包含的词的交集中的元素个数, 与两个节点句包含的词的并集中的元素个数之比. 例如, 节点句 “特朗普/携/夫人/访问/英国” 与节点句 “特朗普/与/首相/共同/接受/记者/采访” 的词的并集中有 11 个词, 但是交集中只有一个词 “特朗普”, 则连接这两个节点的边的权重被初始化为 $\frac{1}{11}$.

2.2 关键子句筛选和词权计算

子句连通图的构建能够帮助我们理顺文章的主要内容, 显然那些描述主要内容的句子应该位于连通图的核心区域内, 即其应该与文章中的很多子句均有关联. 为此, 本文将关键子句的筛选看作是图中节点的排序问题. 通过计算图中节点的中心度 (central degree) [25], 选择排序靠前的 1/3 子句作为关键子句用于构建篇章事件连通图.

本文使用类似 EM 算法的思想对图中节点的权值和边的权值进行更新. 首先使用 TextRank 算法对节点权重进行计算, 这一步相当于 EM 算法中的 E 步. 接着根据计算得到的节点权重, 更新文档中每一个词的词权, 并且依据词权更新图中的边权, 这一步相当于 EM 算法中的 M 步. 之后, 不断迭代执行 E 步和 M 步直到节点权重、边权、词权收敛, 更新步骤见本小节末尾.

TextRank 算法的灵感来源于 Google 提出的网页重要性排序算法 PageRank [26]. 在 PageRank 算法里, 互联网相当于一张巨大而复杂的连通图, 连通图中的节点是现实中的网页. 网页之间通过链接

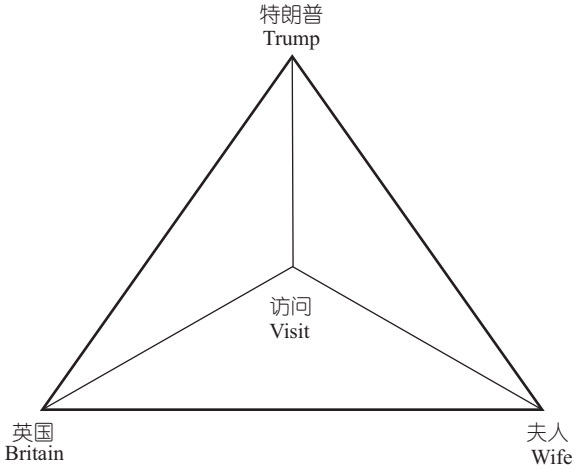

图 1 子句连通图

Figure 1 Sentence-level event graph

互相连通, 即若网页 1 中包含一个指向网页 2 的链接, 则在连通图中代表网页 1 的节点会通过一条有向边与代表网页 2 的节点相连. 直观上认为, 一个网页被越多的其他网页链接, 则这个网页越重要. 映射到连通图中, 若一个节点与越多的其他节点通过边相连, 则证明这个节点越重要. PageRank 算法通过迭代的计算节点权重直至收敛, 来获得对应网页的重要性大小, 迭代公式如下所示:

$$S\left(V_i\right)=(1-d)+d\times\sum_{V_j\in\mathrm{in}(V_i)}\frac{1}{\left|\mathrm{out}\left(V_j\right)\right|}S\left(V_j\right), \tag{2}$$

其中, $S(V_i)$ 为节点 V_i 在一轮迭代后的节点权重. d 为阻尼系数, 是一个超参数, 一般取值为 0.85. 在有向图中, $\mathrm{in}(V_i)$ 代表通过一条边指入节点 i 的节点集合. $|\mathrm{out}(V_j)|$ 代表节点 j 通过一条边指出的节点个数 (出度). 在无向图中, $\mathrm{in}(V_i)$ 代表与节点 i 相连的节点集合, $|\mathrm{out}(V_j)|$ 代表与节点 j 相连的节点个数. $S(V_j)$ 是本轮调整前节点 V_j 的权重.

与 PageRank 相比, TextRank 算法更适用于处理文本, 可用于从文本中提取关键词或者关键句. TextRank 算法首先将一篇文章转化为连通图, 若是要提取关键词则节点代表一个词语. 本文的目的在于提取关键子句, 因此节点代表一个子句. 节点通过两个子句是否具有重合词连接在一起. 依据迭代公式不断调整节点权重直至节点权重收敛. 通过节点权重来衡量子句的重要性. 节点权重越大, 代表对应的子句越重要. 其迭代公式与 PageRank 略有不同, 如下所示:

$$\mathrm{WS}\left(V_i\right)=(1-d)+d\times\sum_{V_j\in\mathrm{in}(V_i)}\frac{w_{ij}}{\sum_{V_k\in\mathrm{out}(V_j)}w_{jk}}\mathrm{WS}\left(V_j\right). \tag{3}$$

相比于 PageRank 利用网页作为节点, 并通过链接连接节点构建有向图, 在 TextRank 算法中是通过滑动窗口将出现在同一窗口中的词语连接起来, 因此 TextRank 算法中的图是无向图. 除此之外, 本文的目的是度量句子的权重以选择关键子句. 故此, 本文对 TextRank 算法做了些许改变, 即以文中的句子作为节点. 如果两个句子共享相同的词语, 则将代表两个句子的节点连接起来. 后续的计算过程即是按照 TextRank 的计算流程. 在式 (3) 中, $\mathrm{WS}(V_i)$ 是节点 V_i 本轮调整后的权重值. 参数 d, $\mathrm{in}(V_i)$ 和 $\mathrm{out}(V_j)$ 的意义和 PageRank 公式类似, 但是由于 TextRank 算法中的图为无向图, 因此其某个节点, 例如 V_i 的链入节点集合, 即 $\mathrm{in}(V_i)$, 和链出节点集合, 即 $\mathrm{out}(V_i)$, 是一样的. 所不同的是 w_{ij} 和 w_{jk},

其中, w_{ij} 代表连接节点 V_i 与节点 V_j 的边的权重, w_{jk} 代表连接节点 V_j 与节点 V_k 的边的权重. 边的权重由节点权重计算得出, 具体计算方法见式 (5).

利用 TextRank 更新节点权重的具体步骤如下:

(1) 初始化: 随机初始化每个节点的权重为 1, 文档中每个词的权重也为 1, 边的权重为边相连的两个节点的重合词的个数与两个节点包含的总词数之比. 具体计算方法见式 (5);

(2) 采用式 (3) 对连通图中的节点权重进行计算;

(3) 根据新的节点权重更新每一个词的权重, 词语 w_i 的权重计算公式为

$$\mathrm{Score}(w_i)=\frac{\sum_{k\in \mathrm{Contain}(w_i)}\mathrm{Score}(V_k)}{\log\left(\frac{\sum_{k\in \mathrm{All}(V)}\mathrm{Score}(V_k)}{\sum_{k\in \mathrm{Contain}(w_i)}\mathrm{Score}(V_k)}\right)\times|\mathrm{All}(V)|}, \tag{4}$$

其中, $\mathrm{Score}(w_i)$ 为单词 w_i 的权重分数, $\mathrm{Contain}(w_i)$ 为包含 w_i 的节点集合, $\mathrm{Score}(V_k)$ 为节点 V_k 的权重分数, $\mathrm{All}(V)$ 为全部节点, $|\cdot|$ 为集合中的元素个数.

(4) 根据更新的词权重, 再次利用式 (5) 计算边的权重;

(5) 迭代 (2)~(4) 步直至收敛;

(6) 获取连通图中的节点权重和词权重.

式 (5) 展示了利用词权计算边权的方法:

$$\mathrm{Score}\left(V_1,V_2\right)=\frac{\sum_{w_i\in \mathrm{lntersection}(V_1,V_2)}\mathrm{Score}\left(w_i\right)}{\sum_{w_i\in \mathrm{Union}(V_1,V_2)}\mathrm{Score}\left(w_i\right)}, \tag{5}$$

其中, $\mathrm{Score}(V_1,V_2)$ 代表连接节点 V_1 和 V_2 的边的权重, $\mathrm{Union}(V_1,V_2)$ 和 $\mathrm{lntersection}(V_1,V_2)$ 的介绍如式 (1) 所示, 这里就不赘述了.

在获得节点权重之后, 按照节点权重由大到小对子句进行排序. 提取排名靠前的 1/3 子句视为核心子句, 作为后续句子级事件提取的对象. 同时, 将得到的节点权重和词权重予以保留以用于初始化后续即将构建的篇章事件连通图中的节点权重和边权重.

2.3 句子级事件多边形的构建

利用 2.2 小节的关键子句抽取方法, 能够获取涵盖文章主要内容的句子. 本小节则以获得的关键子句为基础构建句子级事件多边形用以揭示句子级的事件内容, 为后续篇章事件连通图的构建打下基础.

具体而言, 即是以句子级的事件触发词和事件元素作为篇章事件连通图中的节点. 值得指出的是, 句子级事件的提取结果 (触发词和事件元素) 是为构建篇章事件连通图服务的, 因此句子级事件的提取结果无需非常准确, 但是需要尽可能涵盖丰富的信息, 以供连通图挖掘. 本文采用 Qiu 和 Zhang [27] 提出的 ZORE 模型作为句子级事件的提取方法. ZORE 以三元组模板提取事件, 具体类型为 (事件元素, 关系, 事件元素). 本文仅保留关系标签为 “is”、“de” 和 “动词” 的三元组. 同时, 若关系标签内容为一个动词或动词短语, 则将标签内容作为事件触发词, 关系的头和尾作为事件元素; 若标签内容为 “is” 和 “de”, 则只提取头和尾作为事件元素, 事件触发词用字母 “Z” 代替.

将提取出的事件触发词、事件元素作为节点, 连接成一个具有中心节点的封闭多边形. 比如句子 “特朗普携夫人访问英国” 中提取出事件元素 “特朗普”、“夫人”、“英国”, 提取出事件触发词 “访问”. 首先将 “访问” 与 “特朗普”、“夫人”、“英国” 分别相连, 再将 “特朗普”、“夫人”、“英国” 按照其在句子中出现的顺序依次连接起来. 构成如图 2 所示的句子级事件多边形用以揭示句子级别的事件

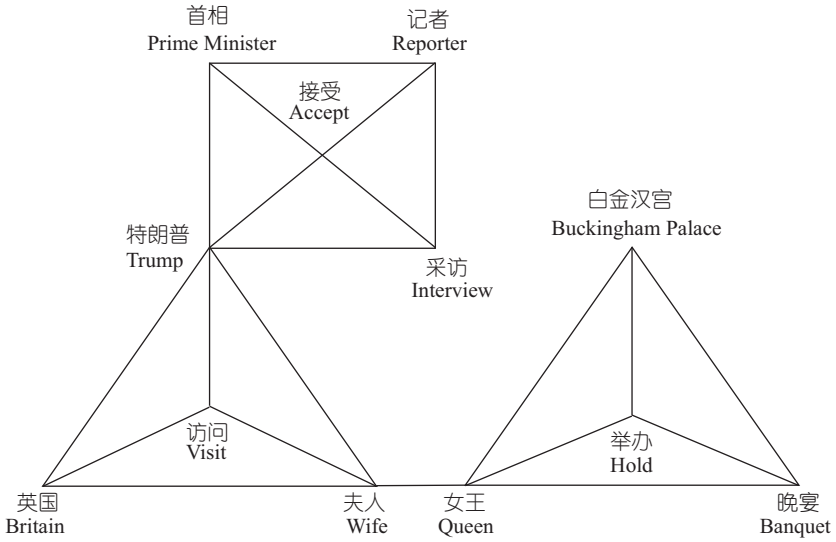

图 2 句子级事件多边形

Figure 2 Sentence-level event polygon

信息. 这样, 整个句子级事件就用一个带有中心节点的封闭多边形表示出来了. 如果一个事件具有 n 个事件元素, 则该事件表示为一个具有中心节点 n 边形.

2.4 篇章事件连通图的构建

利用上文构建的句子级事件多边形为基础即可构建出篇章事件连通图. 由于上文构建的事件多边形是面向单个句子的, 而本文的目的是获得篇章级的事件表示, 因此需要构建跨句子级的事件表示. 本文中, 跨事件的节点链接有两种方式, 第 1 种是如果一个事件元素既出现在一个事件中又出现在另一个事件中, 则这两个事件多边形共用一个节点; 第 2 种是如果两个多边形中有某两个节点对应词语的词向量的余弦相似度超过预先设定的阈值 (阈值设为 0.8, 其设置原因由文献 [28] 给出, 该文献指出, 如果两个词的词向量的余弦相似度高于 0.8, 即可认为这两个词具有类似的语义信息), 则将这两个节点连接在一起. 如下面 3 句话:

- “特朗普携夫人访问英国.”
- “特朗普与首相共同接受记者采访.”
- “女王在白金汉宫举办欢迎晚宴.”

构成的篇章事件连通图如图 3 所示, 其中 “特朗普” 被两个事件多边形所共享, 而 “夫人” 和 “女王” 两个节点对应的词向量的余弦相似度大于 0.8, 故以边连接. 本文采用 Glove 词向量 [29].

2.5 篇章级事件相关度计算

基于构建的篇章事件连通图能够清晰地反映出文章中事件间的联系, 之后即可对事件连通图中的节点排序以从中选出能够代表篇章级事件的节点. 一篇文章的核心事件应该和文中描述的其他事件有或多或少的联系. 故此, 如果以图的形式表示, 那么事件连通图中的核心事件节点应该和其他事件具有大量的联系. 从节点中心度度量的角度去看, 图中节点的权值能够代表该节点在事件连通图中的重要性, 即是否能够作为核心事件的代表. 本文仍采用 TextRank 算法计算图中节点的权重. 在获取节点权重后, 本文根据权重大小对连通图中的节点进行 3 类排序: (1) 把文档中所有的命名实体按照权重

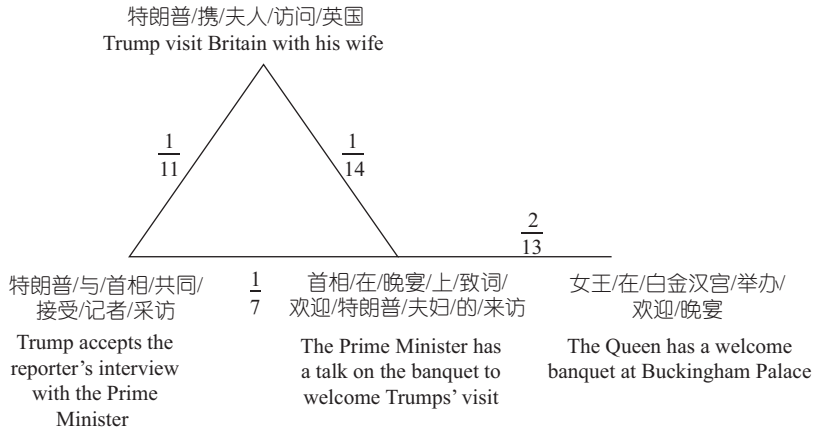

图 3　篇章级事件连通图

Figure 3　Passage-level event connection graph

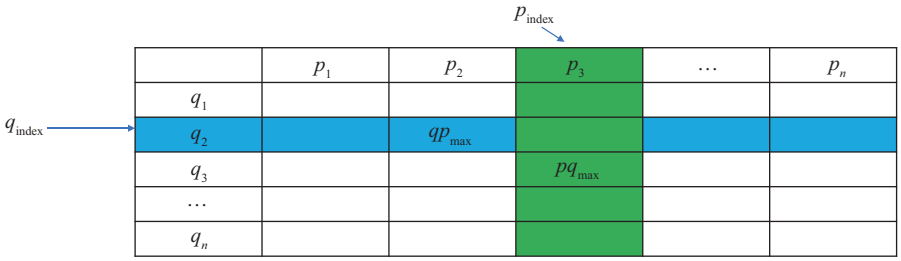

图 4　(网络版彩图) 余弦相似度矩阵图

Figure 4　(Color online) Matrix of cosine similarity

大小进行排序; (2) 把文档中所有的触发词按照权重大小进行排序; (3) 把文档中所有的非实体名词按照权重大小进行排序.

将每个排行榜上的前 5 个词提取出来作为篇章级事件的代表. 将从两篇文档 P 和 Q 中提取出的词语按照其在文档中出现的先后顺序排列成两个关键事件词序列 $p_1, p_2, p_3, \ldots, p_n$ 与 $q_1, q_2, q_3, \ldots, q_n$. 将 $p_1, p_2, p_3, \ldots, p_n$ 中的每一个词与 $q_1, q_2, q_3, \ldots, q_n$ 中的每一个词两两之间计算词向量的余弦相似度, 并组成余弦相似度矩阵, 如图 4 所示. 其中, $qp_{\max}$ 是 q_2 所在行的余弦相似度的最大值, $pq_{\max}$ 是 p_3 所在列的余弦相似度的最大值. 找出每一行的 $qp_{\max}$ 和每一列的 $pq_{\max}$, 将所有的 $qp_{\max}$ 求和得到每行最大值的和 q_{sum}, 将所有的 $pq_{\max}$ 求和得到每列最大值的和 $p_{\max}$. 根据式 (6) 计算出加权平均值 avg.

$$\text{avg} = \frac{q_{\text{sum}}/q_{\text{size}} + p_{\text{sum}}/p_{\text{size}}}{2}, \tag{6}$$

其中, q_{size} 代表由文档 Q 构建而成的篇章事件连通图中节点的个数, p_{size} 则代表由文档 P 构建而成的篇章事件连通图中节点的个数. 式 (6) 中, $q_{\text{sum}}/q_{\text{size}}$ 代表了文档 Q 中的每个词相对于文档 P 的平均相似度, 而 $P_{\text{sum}}/P_{\text{size}}$ 代表了文档 P 中的每个词相对于文档 Q 的平均相似度.

3 实验与结果分析

3.1 数据集与评估方法

本文采用了两类语料用于验证本文提出方法的有效性. 一类语料来源于标准数据集, 一类语料来源于手动标注. 由于当前缺少针对中文的文本相关度评测数据集, 因此本文选择新闻文本分类语料作为标准评测数据集. 文本分类的目的是按照给定的类别将相似的文本归类, 而对于新闻来说, 其核心在于其描述的新闻事件, 故而, 针对新闻的文本分类均是按照新闻描述事件的不同而进行归类. 基于此, 本文选择新闻文本分类语料来衡量本文所提出的文本相关度计算方法的性能. 本文选用搜狗新闻数据集和今日头条新闻数据集作为测试数据集, 其中搜狗数据集共包含 18 个类别, 今日头条数据集共包含 15 个类别. 本文分别从以上数据集中随机选择类别, 然后从类别内随机抽取两篇新闻文本作为相关文本对, 依据以上方案从每个数据集中随机抽取 2000 对新闻文本, 之后再通过随机组合不同的类别 (即随机抽取不同的类别, 并从中各抽取一篇文档) 抽取 2000 对新闻文本作为不相关文本对. 这 4000 对文本中, 分别选择 1000 对相似文本和 1000 对不相似文本作为验证集用来选择合适的相似度度量阈值, 剩下的 2000 对文本作为测试集.

上述的文本分类语料通过两篇文本是否位于同一类别来衡量两篇文本是否相关, 这种衡量方式过于粗糙. 因为即使两篇文档位于相同的类别, 他们也很可能描述了不同的事件. 例如, 两篇来源于体育类别的文档, 可能一篇描述足球事件, 一篇描述篮球事件. 基于此, 本文期望人工构建细粒度的评测语料来更加准确地评价不同文本相关度计算方法的性能.

针对人工构建的语料, 本文从天天快报网站爬取 1000 篇体育领域的新闻作为实验语料. 从 1000 篇新闻中随机抽取 200 篇新闻, 人为标注 200 篇新闻的核心事件, 共分为 27 个类别标签, 如 “足球”、“篮球”、“转会”、“球员交易” 等. 按照类别标签对 200 篇新闻进行划分, 将位于同一类别内的新闻两两组合, 之后随机抽取 2000 对新闻对作为数据集. 数据集中的任一对新闻均具有相同的类别标签. 由人工判断两篇类别标签相同的新闻是否相关, 如相关则将两篇新闻组成的新闻对标注为相关新闻对, 否则标注为不相关新闻对. 共招募了 3 名论文作者所在学院的大三本科生对新闻对是否相关给予投票, 取票数最高的结果作为标注结果. 采用这种方案构建数据集的原因在于: 我们期望本文提出的算法不仅能够获取文本中的核心事件, 并且能够利用核心事件来度量不同文本之间描述的核心事件的相关度. 如果两篇新闻分属于不同的类别标签, 说明这两篇新闻描述的事件之间没有任何关联, 那么区分这两篇新闻的相关度对于实际的应用, 例如新闻推荐, 没有任何意义. 故此, 我们在构造数据集的时候期望增大数据集的难度, 即期望能够确定隶属于同一类别标签下的新闻对的相关性. 将标注的 2000 对新闻对中的 1000 对作为验证集用来选择合适的相关度度量阈值, 剩下的 1000 对作为测试集.

本文将准确率 P、召回率 R 和 $F1$ 值作为评估标准用来评估本文所提方法的性能.

准确率 P. 识别出的所有正确的相关新闻对的个数, 与识别出的所有相关新闻对的个数的比值.

召回率 R. 识别出的所有正确的相关新闻对的个数, 与相关新闻对的总个数的比值.

$F1$ 值. 由准确率 P 和召回率 R 共同计算得出. 由于 $F1$ 值是经典的评估方法, 这里就不做过多的介绍了, 具体的计算公式请参见文献 [30].

3.2 数据集处理

将数据集中的新闻使用哈尔滨工业大学社会计算与信息检索研究中心研发的语言技术平台 LTP 进行命名实体识别、依存句法分析 [31], 使用 Stanford University 研发的语言分析器进行短语结构句法分析 [32]. 利用句法树结构, 融合依存句法分析和短语结构句法分析结果, 依照文中 2.1∼2.3 小节介绍

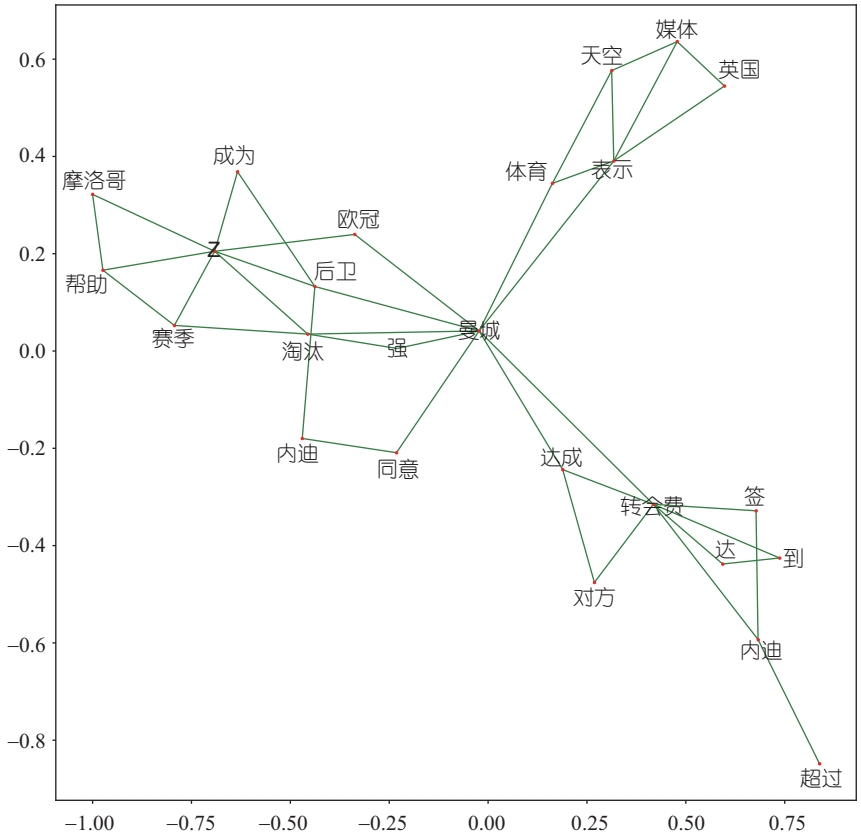

图 5　(网络版彩图) 篇章事件连通图示例图

Figure 5　(Color online) One example of passage-level event connection graph

的方法进行事件触发词和事件元素的识别.

3.3　篇章事件连通图样例

将文档中的每一个句子级事件的事件触发词作为中心节点, 事件元素作为多边形顶点构造事件多边形, 之后即可按照 2.4 小节提出的方法将事件多边形的节点进行连接、融合, 得到篇章事件连通图. 图 5 为一篇新闻的篇章事件连通图样例 (横纵坐标代表缩放程度), 这篇新闻对应的事件是 “曼城 5200 万英镑引入左后卫门迪”[1)].

3.4　篇章级事件相关度的计算结果样例分析

按照 2.5 小节介绍的方法, 首先采用 TextRank 对图 5 中的节点权重进行计算并排序, 之后按照下述 3 个指标提取词语作为篇章级事件的代表. (1) 权值排名前 5 的触发词; (2) 权值排名前 5 的命名实体; (3) 权值排名前 5 的词语.

由于文章长度的限制, 新闻 “曼城同意 5200 万英镑引入左后卫门迪” 只提取出排名前 3 的词语, 获得的 3 个指标分别为: (1) 表示、Z、达成; (2) 曼城、门迪、英国; (3) 达成、转会费、Z. 由提取出的指标结果可见, 这些提取到的词语能够反映出该篇新闻的核心事件是有关 “球员转会费达成” 一事.

1) http://kuaibao.qq.com/s/20170722C05FBM00.

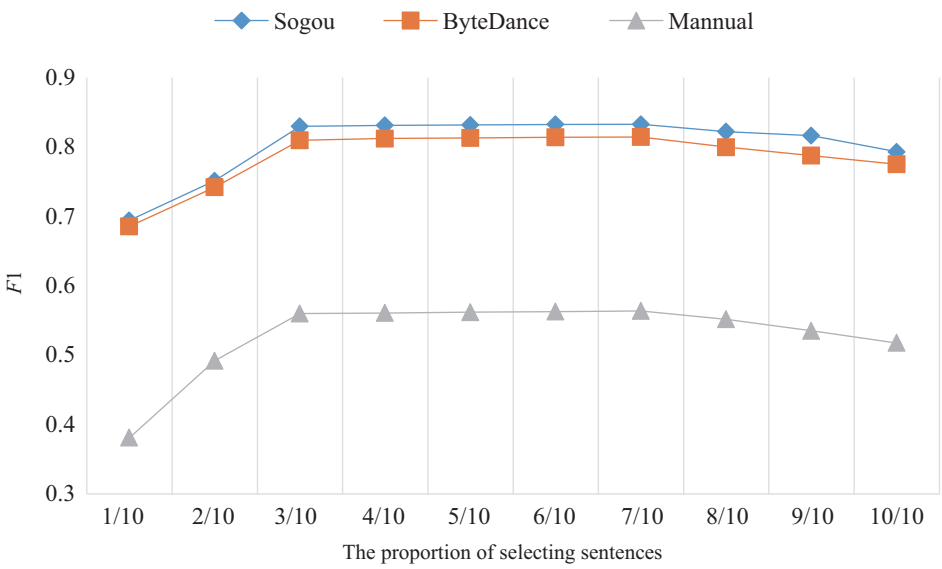

图 6 (网络版彩图) 本文所提方法在选择不同比例子句下的性能图

Figure 6 (Color online) Performance graph on different selecting proportions of sentences

3.5 参数分析

本文提出的篇章事件相关度计算方法可分为 4 个阶段, 分别为: (1) 关键子句筛选, (2) 句子级事件多边形构建, (3) 篇章级事件连通图构建, (4) 篇章级事件相关度计算. 其中, 在关键子句筛选步骤中需要设置阈值来选择核心子句以构建句子级事件多边形, 同样在篇章级事件相关度计算步骤中需要根据由 TextRank 算法计算出的词权, 设置阈值以分别选出排序在前列的命名实体、触发词和非命名实体用以计算文本相关度. 本文分别设置上述的阈值为 1/3 和 5, 即分别选择排序在前 1/3 的核心子句来构建句子级事件多边形和选择排序在前 5 的词语作为篇章级事件的代表以计算文本间的相关度.

本文在构建测试数据集时从数据集中筛选出一半的文本对作为验证集以确定本文所提方法中需要预先设定的阈值. 图 6 和 7 显示了不同的阈值设置下本文所提出的文本相关度计算方法在测试语料上的效果 (以 $F1$ 值度量). 其中, 图 6 显示了在根据权值选择不同比例的子句 (从 1/10 至 1/1) 下, 本文提出的文本相关度计算方法的性能, 相对的, 图 7 则显示了在选择不同个数的词语 (从 1 至 10) 时, 本文提出的文本相关度计算方法的性能.

由图 6 可见, 在不同的语料集下 (搜狗语料、今日头条语料、人工构建语料), 其性能曲线几乎是类似的, 即随着子句比例的提升, 本文提出的文本相关度计算方法的性能呈上升趋势, 而当选择子句的比例接近 1/3 时, 其性能趋近平衡. 但是, 随着子句比例的提升 (接近 4/5), 其性能又呈现下降趋势. 产生此种情况的原因在于, 本文依靠 TextRank 算法为文本中的句子赋予权重, 权重越大, 表明句子的重要性越大, 即越能够代表文本的主题. 故此, 随着句子选择比例的提升, 越来越多能够反映文本主题的句子被采纳用于构建后续的句子级事件多边形和篇章事件连通图. 故而, 构建的图形能够非常好地揭示文本的核心事件. 当选择的句子达到一定比例时, 即 1/3, 大部分与文本核心事件相关的句子均已被纳入来构建篇章事件连通图. 同时, 对于选择的不相关的句子, 会通过后续的事件连通图中节点权值的计算予以过滤 (对不相关句子中的词语赋予较小的权值). 但是, 随着选择句子比例的提升, 例如全部选择, 此时大量的无关信息的涌入会减弱利用节点权值过滤无关信息的效果, 故而文本相关度计算结果的准确率会有所降低. 同时需要指出的是, 本文提出的文本相关度计算方法在搜狗语料和今日

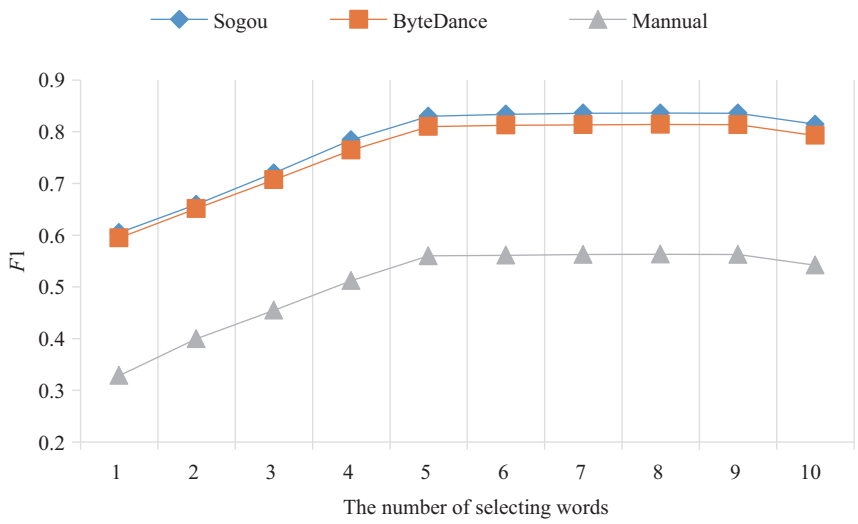

图 7　(网络版彩图) 本文所提方法在选择不同数量词语下的性能图

Figure 7　(Color online) Performance graph on different selecting numbers of words

头条语料上获得的准确率较高, 而在人工构建的语料上的准确率较低. 其原因来源于语料构建的差异. 其中, 搜狗语料和今日头条语料仅依靠两篇新闻文本是否归属于同一类别而确定两篇新闻文本是否相关. 文本相关度计算方法是很容易区分出两篇文本是否归属于某一类别下的 (例如, 是否都是描述体育或者金融类事件的新闻), 而人工构建的语料则要细致很多, 其目标在于更加细粒度地区分文本之间的相关性, 即判断两篇来自同一类别的新闻文本是否描述了类似的事件. 其语料的难度造成了文本相关度计算方法的性能的下降.

图 7 中的相关度计算结果趋势和图 6 类似, 即同样随着选择词语数量的增加, 其性能呈上升趋势, 而当达到一定数值时 (接近 9), 其性能略有下降. 造成此结果的原因和图 6 类似, 即随着选择词语数量的提升, 越来越多的能够代表文本核心事件的词语被选择作为特征用于计算文本之间的相关度. 故而, 其文本相关度计算结果的准确率会随着选择词语数量的增加而逐渐提高. 当选择的词语特征达到一定数量时, 即超过 5 个, 大部分与文本核心事件相关的词语已经能够被很好的覆盖. 此种情况造成文本相关度计算结果的准确率不会随着选择词语数量的增加而持续提升. 然而, 随着选择词语数量的增加, 例如达到 10, 此时大量代表无关信息的词语的加入则会造成性能曲线的下降.

3.6　对比实验结果与分析

对测试集中的每个新闻对, 可按照 2.5 小节介绍的方法分别计算每个指标包含的词语的词向量两两之间的余弦相似度, 取最大值加权平均后即可得到两篇文档间的篇章级事件相似度. 当相似度大于阈值时 (阈值设定由验证集给出), 则判定两篇新闻是相关的, 否则为不相关. 本文选择了 4 类方法用于衡量本文所提出的文本相关度计算方法的性能.

第 1 类方法为无监督的对比方法 (前 8 种文本相关度计算方法). 本文所提出的方法即为一种无监督的文本相关度计算方法. 采用无监督的方法计算文本相关度的原因在于当前缺少足够地用于衡量文本间描述的事件是否相关的大规模训练语料. 故此, 为充分衡量本文所提方法的有效性, 共设计了 8 种无监督的对比实验, 实验中充分地考虑了文本中词的统计信息 (词频、词权重、词跨度、互信息) 和语义信息 (词的嵌入向量表示) 的各种组合. 在对比实验中用来判断文档是否相关的阈值均由验证集给

出, 具体做法参见对比方法介绍之后的说明.

第 2 类方法为有监督的对比方法 (中间的 3 种文本相关度计算方法). 当前针对有监督的文本间相似度或相关度计算方法均集中于句子级别, 即用来判断句子, 尤其是问句, 之间的相似度. 对于有监督的方法需要大规模的训练语料, 而本文所提方法的背景包括本文所人工构建的语料都是面向语料缺失的情况, 因此缺少必要的训练语料用于训练有监督的方法. 故此, 本文首先在标准数据集上测试了有监督的对比方法. 即利用构建验证集和测试集的方案, 构建了两万对新闻文本训练语料, 其中一万对文本为相关文本, 另外一万对为不相关文本, 之后利用此训练语料训练有监督的对比方法. 在人工构建的小规模数据集上, 本文以上述通过训练集训练出来的方法参数为基础, 采用人工构建数据集中的验证集中的 1000 对文本对方法中的参数做微调, 之后利用得到的新参数来计算文本间的相关度.

第 3 类方法为预训练语言模型 (之后的 2 种文本相关度计算方法). 当前的预训练语言模型, 例如 BERT, 在很多自然语言处理任务上都证明了其出色的性能. 其通过大规模的无标签训练数据和深层网络能够将大量预训练文本中的语法或语义等先验知识固化到其参数之内, 故此, 在很多下游任务上获得了较好的性能. 本文也将预训练模型作为测试方法之一, 仿照为有监督方法构建数据集的方式构建训练集用于对预训练模型进行微调, 同时利用验证集选择合适的迭代次数. 由于预训练模型有输入长度的限制 (512 个单词), 而本文采用的语料中无论是标准的文本分类语料还是人工构建的小规模语料都是长文本, 长度大多大于 512 个单词. 为此, 本论文采用滑动窗口方式, 窗口长度为 512 (包括一个 [CLS] 标签). 利用滑动窗口从头到尾对输入文档扫描, 假设两篇文档为 P 和 Q, 以文档 P 为例, 利用滑动窗口, 由 P 可获得 m 个窗口, 分别将这 m 个窗口中每个窗口对应的 [CLS] 标签作为当前窗口的表示, 然后将所有窗口的 [CLS] 表示求平均作为文档的表示. 文档 Q 也采用同样的方法. 之后利用一个全连接层以文档 P 和文档 Q 的表示的拼接作为输入, 并利用 softmax 获得两篇文档之间相关度数值.

第 4 类方法为基于文摘的方法 (最后的 3 种文本相关度计算方法). 文摘方法的核心在于抽取出能够代表文章主题的句子. 事实上, 我们也可以将文摘句作为文本的代表, 然后计算文摘句之间的相关度作为文本之间的相关度. 为此, 本文选择了相应的文摘抽取算法从新闻文本中抽取文摘句. 由于文摘句中会存在一些无用信息, 例如停用词、形容词等, 其对计算文本间的相关度没有任何帮助. 故此, 本文对抽取出的文摘句利用 LTP 语言技术平台分词去除停用词后, 将保留的词语作为特征, 利用这些特征对文档向量化后采用余弦相似度来计算文本之间的相关度.

表 1∼3 中分别列出了本文所提方法和上述 3 类方法的对比实验结果. 针对表 2 和 3 中的有监督的对比方法和基于文摘的对比方法, 我们使用其在对应论文中的简称称呼它们.

基于词频 – 逆文档频率的文本相关度计算 —— TF/IDF. 对测试语料中的新闻使用 LTP 进行分词、去重, 以形成测试语料的词表, 对词表中的每一个词, 计算该词相对于每一篇新闻的词频 – 逆文档频率, 得到词语 – 篇章矩阵. 词频 – 逆文档频率 $P_{i,j}$ 的计算公式如下:

$$P_{i,j} = \frac{n_{i,j}}{\sum_i n_{i,j}} \times \log \frac{N}{n(i)}, \tag{7}$$

其中, $n_{i,j}$ 为词语 i 在篇章 j 中出现的次数, $n(i)$ 为词语 i 在整个语料库中出现的总次数, N 是语料库中文档的总数. 对每一篇新闻中的词语按照词频 – 逆文档频率由大到小进行排序, 取前 10 个词组成词集合, 计算两篇文档的词集合的重合程度, 若重合程度超过阈值则判定两篇文档相关, 反之为不相关.

基于词频 – 逆文档频率及余弦相似度的文本相关度计算 —— TF/IDF+Cosine. 在对比实验一中获取了按照词频 – 逆文档频率排序前十的词语集合后, 计算两篇文档对应的两个词集合中的词语

表 1 不同的无监督方法的测试结果

Table 1 Results of different unsupervised methods[a)]

Method	Data set	P (%)	R (%)	$F1$
TF/IDF	Sogou	84.67	55.43	0.67
	ByteDance	81.66	55.91	0.65
	Mannual	50.79	10.66	0.17
TF/IDF+Cosine	Sogou	83.37	21.77	0.35
	ByteDance	79.44	20.59	0.32
	Mannual	**77.88**	2.66	0.05
TF+Cosine	Sogou	84.44	64.33	0.73
	ByteDance	82.17	65.12	0.73
	Mannual	63.43	38.01	0.42
MI+Cosine	Sogou	85.29	68.25	0.76
	ByteDance	83.71	66.73	0.74
	Mannual	69.36	35.18	0.46
WS+Cosine	Sogou	86.45	66.29	0.75
	ByteDance	**86.41**	65.11	0.74
	Mannual	70.41	33.97	0.45
FE+Cosine	Sogou	**87.25**	63.33	0.73
	ByteDance	86.31	59.28	0.70
	Mannual	74.67	17.26	0.28
TR+Cosine	Sogou	84.67	78.52	0.81
	ByteDance	82.79	**77.31**	0.80
	Mannual	67.13	**46.13**	0.54
EN+Graph	Sogou	81.22	72.19	0.76
	ByteDance	77.65	73.15	0.75
	Mannual	33.15	42.67	0.37
Ours	Sogou	86.54	**79.29**	**0.83**
	ByteDance	85.77	76.26	**0.81**
	Mannual	75.89	43.97	**0.56**

a) The value of black bold indicates the maximum value per column.

两两之间的词向量的余弦相似度, 将所有相似度求和后取平均值并设置阈值, 若计算结果大于阈值则为相关, 反之为不相关.

基于词频及余弦相似度的文本相关度计算 —— TF+Cosine. 对文档中的词语按照词频大小由大到小排序, 提取出排名在前 20% 的词作为关键词. 计算两篇文档对应的两个词集合中的词语两两之间的词向量的余弦相似度, 将所有相似度求和后取平均值并设置阈值, 若计算结果大于阈值则为相关, 反之为不相关.

基于互信息及余弦相似度的文本相关度计算 —— MI+Cosine. 对文档中的每个词语计算其互信息, 并按照互信息大小由大到小排序, 提取出排名在前 20% 的词构成词集合. 计算两篇文档对应的两个词集合中的词语两两之间的词向量的余弦相似度, 将所有相似度求和后取平均值并设置阈值, 若

表 2 不同的有监督方法的测试结果

Table 2 Results of different supervised methods[a)]

Method	Data set	P (%)	R (%)	$F1$
MwAN	Sogou	84.24	78.15	0.81
	ByteDance	84.56	75.93	0.80
	Mannual	73.25	30.34	0.43
DIIN	Sogou	**88.57**	76.83	0.82
	ByteDance	85.36	**77.22**	**0.82**
	Mannual	**76.15**	33.57	0.47
DRCN	Sogou	87.82	78.44	**0.84**
	ByteDance	84.51	76.78	0.81
	Mannual	71.44	38.67	0.50
Ours	Sogou	86.54	**79.29**	0.83
	ByteDance	**85.77**	76.26	0.81
	Mannual	75.89	**43.97**	**0.56**

a) The value of black bold indicates the maximum value per column.

表 3 不同的预训练模型的测试结果

Table 3 Results of different pre-training methods[a)]

Method	Data set	P (%)	R (%)	$F1$
BERT	Sogou	86.79	**83.28**	0.85
	ByteDance	86.33	76.29	0.81
	Mannual	74.29	37.68	0.50
BERT-wwm	Sogou	**93.78**	82.88	**0.88**
	ByteDance	**92.13**	**78.89**	**0.85**
	Mannual	**77.29**	39.17	0.52
Ours	Sogou	86.54	79.29	0.83
	ByteDance	85.77	76.26	0.81
	Mannual	75.89	**43.97**	**0.56**

a) The value of black bold indicates the maximum value per column.

计算结果大于阈值则为相关, 反之为不相关.

基于词跨度及余弦相似度的文本相关度计算 ——WS+Cosine. 对文档中的词语按照词跨度大小由大到小排序, 词跨度的计算方法参见文献 [33], 提取出排名在前 20% 的词构成词集合. 计算两篇文档对应的两个词集合中的词语两两之间的词向量的余弦相似度, 将所有相似度求和后取平均值并设置阈值, 若计算结果大于阈值则为相关, 反之为不相关.

基于特征工程及余弦相似度的文本相关度计算 —— FE+Cosine. 对文档中的词语计算词频、TF/IDF 值、互信息、词跨度 4 个指标, 并为每个指标赋予相同的权重. 将 4 个指标值加权求和作为每个词语最终的分数. 按分数由大到小排序, 提取出排名在前 20% 的词构成词集合. 计算两篇文档对应的两个词集合中的词语两两之间的词向量的余弦相似度, 将所有相似度求和后取平均值并设置阈值, 若计算结果大于阈值则为相关, 反之为不相关.

基于 TextRank 及余弦相似度的文本相关度计算 —— TR+Cosine. 使用 TextRank 算法对文

档中每一个词的词权进行计算. 对文档中的词语按照词权由大到小排序, 提取出排名在前 20% 的词构成词集合. 计算两篇文档对应的两个词集合中的词语两两之间的词向量的余弦相似度, 将所有相似度求和后取平均值并设置阈值, 若计算结果大于阈值则为相关, 反之为不相关.

基于以事件为节点的连通图的文本相关度计算 —— EN+Graph. 该对比实验与本文提出的方法有相似之处. 其以提取到的句子级事件为节点构建连通图 (即每个句子级事件作为图中的一个节点), 以两个事件中的触发词的余弦相似度与事件中所有命名实体两两之间的余弦相似度之和的平均值作为节点相似度. 若节点相似度大于阈值, 则将节点连接起来, 否则两个节点相互独立. 构成连通图后采用 TextRank 计算每个节点的权重并按照权重由大到小排序. 分别提取两篇文档中权重排在前 3 的事件节点, 将所有事件节点中包含的词语取并集作为该篇文档的代表. 计算两篇文档对应的两个词集合中的词语两两之间的词向量的余弦相似度, 将所有相似度求和后取平均值并设置阈值, 若计算结果大于阈值则为相关, 反之为不相关.

基于多路注意力机制的文本相关度计算 —— MwAN [34]**.** 该方法首先分别将需要计算相关度的两篇文档输入到不同的 LSTM 中, 之后采用 4 种方式计算两篇文档间的注意力值. 将得到的注意力值经过组合并通过多层 LSTM 后即可获得文档的更新表示, 此更新表示能够很好地建模文档之间的内容交互性, 最后利用 softmax 层进行分类即可获得两篇文档的相关度值. 将所有相关度值求和后取平均值并设置阈值, 若计算结果大于阈值则为相关, 反之为不相关.

基于交互的文本相关度计算 —— DIIN [35]**.** 该方法首先使用 Bi-LSTM 对输入的两篇文档进行建模, 然后依次对输入文档中的词表示进行点积以获得具有交互信息的张量, 再使用 CNN 对获得的张量进行信息提取以分别得到两篇文档的表示矩阵, 最后同样利用 softmax 层进行分类以获得两篇文档的相关度值. 将所有相关度值求和后取平均值并设置阈值, 若计算结果大于阈值则为相关, 反之为不相关.

基于语义的文本相关度计算 —— DRCN [36]**.** 该方法首先获得输入的两篇文档的词向量嵌入表示, 然后利用融合注意力机制的堆叠 RNN 对文档进行深入语义编码表示. 此种编码会获得较长的表示向量从而干扰了后续的相关度计算. 为此, 该方法在编码部分后增加了一个自编码器用来压缩向量, 此压缩向量可看作是对文档的语义表示. 利用此表示过 softmax 分类层以获得两篇文档的相关度值. 将所有相关度值求和后取平均值并设置阈值, 若计算结果大于阈值则为相关, 反之为不相关.

标准的中文预训练语言模型 —— BERT (Chinese) [37]**.** BERT-base (Chinese) 是 Google 公司提出的中文预训练模型. 其通过大规模的无标签训练数据和深层网络能够将大量预训练文本中的语法或语义等先验知识固化到其参数之内, 在很多下游任务上获得了较好的性能. 参照前述针对预训练方法的介绍, 本文利用滑动窗口扫描文本, 并获得每个窗口内文本片段的表示, 将所有片段的表示取平均作为文本的表示. 之后, 将两篇文本的表示拼接后过全连接层, 再利用 softmax 以获得两篇文档之间的相关度值. 将所有相关度值求和后取平均值并设置阈值, 若计算结果大于阈值则为相关, 反之为不相关.

基于全词 mask 的中文预训练语言模型 —— BERT-wwm (Chinese) [38]**.** BERT-wwm (Chinese) 是哈尔滨工业大学讯飞联合实验室发布的中文预训练模型. 该模型基于 BERT 模型, 但是在模型中考虑了中文的分词特性, 做了全词掩码操作. BERT-wwm 的输入和 BERT 是一样的, 因此针对文档的处理也采用同样的方案. 考虑到参数的可比性, 本文采用的预训练模型均为 base 版本.

基于隐变量的文本相关度计算 —— Extract [39]**.** 该方法利用 Bi-LSTM 依次对输入文档的句子进行编码, 在编码时设置了额外的隐变量以指示编码的句子是否可作为文摘的一部分. 将句子的编码进行叠加后作为文档表示, 并和句子表示进行拼接以使句子的表示获得全局文档信息. 之后利用全

连接层将文档中的所有句子映射到一个 n 维的向量上 (n 代表句子的个数), 每个维度上的值 (0 或 1) 即代表了句子是否被选择作为文摘句的结果. 以得到的文摘为基础, 利用语言技术平台 LTP 分词去除停用词后, 将保留的词语作为文本相关度计算的特征, 利用这些特征对文档向量化后采用余弦相似度计算文档之间的相关度. 将所有相关度值求和后取平均值并设置阈值, 若计算结果大于阈值则为相关, 反之为不相关. 后续基于文摘的文本相关度计算方法均依此处理.

基于联合编码的文本相关度计算 —— NEUSUM [40]**.** 该方法利用端到端的编码解码架构. 其中在编码端将文档切分成句乃至词, 之后经 LSTM 层后获得文档中每个句子的表示. 在解码端利用新的 LSTM, 以此 LSTM 的隐层对编码层输出的每个句子计算注意力值, 以注意力值最大的句子作为当前输出的最优文摘句, 并依此步骤去选择下一句文摘句, 直至算法收敛.

基于门控机制的文本相关度计算 —— SWAP-NET [41]**.** 该方法同样采用端到端的编码解码架构. 与之前介绍的 NEUSUM 类似, 其编码端同样使用 LSTM 建模输入文档, 并利用注意力引导解码端文摘句的选择. 所不同的是在解码端引入了 Pointer Network, 其文摘句的选择由注意力值和 Pointer Network 共同控制, 取最大概率的句子作为当前输出的最优文摘句, 并依此步骤去选择下一句文摘句, 直至算法收敛. 该方法引入门控机制用于调整两方面数值的叠加.

通过验证集可将本文提出的方法中用于判断文档间是否相关的阈值设置为 0.4. 其阈值设置的方案为: 先将阈值分别设置为 $0.1, 0.2, 0.3, \ldots, 1.0$ 等 10 个值, 之后利用每个阈值计算本文提出的方法在验证集上的 $F1$ 值, 然后从中选择最大的 $F1$ 值对应的阈值作为最终设置的阈值. 其他对比方法均采用此方案设置相应的阈值.

由表 1 可见, 在设置的阈值下, 本文所提方法在标准数据集上和人工标注的数据集上均获得了优于 8 种对比方法的计算结果, 即该方法能够找到准确率和召回率之间的最佳折中值, 从而得到最合理的相关度判定结果. 其中, 对比方法 1 通过 TF/IDF 值发现了对文章贡献最大的若干词语, 但是这种方法忽略了事件之间的关联性, 提取出的词语可能全部是事件元素或者全部是事件触发词, 即使是相似的新闻, 在用词上也会有差别, 因此对比方法 1 的准确率和召回率均较低. 对比方法 2∼7 在提取出关键词的基础上引入了余弦相似度, 因此能够在一定程度上发现由于相似用词而引起的事件相关性, 但是此类方法穷举了所有可能的词对, 也引入了大量的噪声. 对比方法 8 能够找出权值最高的事件, 但是这种表示事件的方式过于粗粒度, 两篇围绕同一事件的新闻, 其中一篇新闻提取出的权值最高的事件可能是 “某某球员进球得分”, 另一篇新闻提取出的可能是 “某某球员犯规”, 这种方式归根结底还是通过句子级事件去衡量篇章级事件的相关性, 故此其准确率远低于本文提出的方法. 这也从侧面上说明, 本文提出的方法通过构建细粒度的篇章事件连通图, 能够全面地揭示出文章中陈述的事件内容和事件间的联系, 所提取出的词语能够作为篇章级事件的代表, 因此得到了最优的篇章级事件相关度计算结果.

由表 2 可见, 本文提出的文本相关度计算方法在基于文本分类的标准数据集上取得了和有监督的方法类似的效果, 但是在人工标注的数据集上, 本文提出的方法获得了远高于有监督的方法的性能. 其原因在于, 传统的有监督的文本相关度计算方法均是基于句子级的, 其输入是两个句子 (通常为问句), 然后通过神经网络或其他表示模型编码句子表示. 不同的表示方法决定了建模句子的粒度和深度, 从而影响了判断文本是否相关的结果. 例如, 采用了注意力机制和语言模型的句子建模方法, 如 DRCN, 获得的计算结果要略优于不使用这些模型的方法. 然而, 相对于文本, 句子的长度较短和其覆盖的含义相对集中, 这使得上述基于句子级的有监督方法均力求获得句子的深度表示, 而忽略了不相关信息带来的影响. 这种建模方法在篇章级上则变得不适用起来, 因为文本包含了丰富的信息, 且其中会有很多与主题或核心事件无关的无用信息. 对这些信息进行建模会妨碍到最终的文本相关度计算结果的准

表 4　不同的基于文摘方法的测试结果

Table 4　Results of different abstract extraction based methods[a)]

Method	Data set	P (%)	R (%)	$F1$
Extract	Sogou	85.13	78.46	0.82
	ByteDance	83.81	**77.59**	0.81
	Mannual	74.31	33.26	0.46
NEUSUM	Sogou	83.15	76.11	0.79
	ByteDance	82.55	75.31	0.79
	Mannual	75.44	36.48	0.49
SWAP-NET	Sogou	**86.59**	**79.37**	**0.83**
	ByteDance	85.67	75.93	0.81
	Mannual	74.64	39.50	0.52
Ours	Sogou	86.54	79.19	0.83
	ByteDance	**85.77**	76.16	**0.81**
	Mannual	**75.89**	**43.97**	**0.56**

a) The value of black bold indicates the maximum value per column.

确性. 在标准数据集上, 由于判定文本是否相关的粒度较粗, 所以这些不相关的无用信息不会妨碍到基于句子级的有监督方法的计算结果. 而当判定相关的粒度变细时, 即在人工标注的数据集上, 这种基于句子级的有监督方法会引入大量的无用信息从而降低了对文档建模的有效性. 相对的, 本文提出的文本相关度计算方法一是通过选择关键子句, 二是通过构建篇章事件连通图来抽取与篇章级事件相关的特征. 这些方法保证了选择的特征能够有效地反映出文本的核心事件. 基于此, 本文提出的文本相关度计算方法的性能要远好于有监督的相关度计算方法.

由表 3 可见, 与表 2 的结果类似, 预训练模型在基于文本分类的标准数据集上获得了略高于本文所提方法的结果, 且其性能要优于表 2 中的有监督对比方法. 其原因在于预训练模型通过大规模的无标签训练数据和深层网络能够将大量预训练文本中的语法或语义等先验知识固化到其参数之内, 故此利用预训练模型可以很好地建模文本的表示. 基于文本分类的标准数据集一是具有充足的训练语料, 二是任务的难度相对较低 (因为用来判定文本是否相关的粒度较粗), 故而预训练模型获得了最优的结果. 但是, 由于本文处理的文本长度较长, 而预训练模型只能对固定长度的文本进行处理, 因此, 预训练模型也只能较好地建模文本片段的表示, 而无法做到文档全局的有效建模. 故而, 其在标准数据集上获得的结果低于其在句子级任务上的结果 [37]. BERT-wwm 由于采用了更加适合于中文的全词掩码, 因此其效果优于标准的 BERT 模型. 在人工标注的数据集上, 本文提出的方法获得了高于预训练模型的性能. 其原因和表 2 类似. 预训练模型由于输入长度的限制, 无法做到文档全局的有效建模, 即无法从篇章级别获取文本的有效表示, 而这些全局特征对于从篇章级别判定文本是否相关是非常重要的. 在标准数据集上, 由于判定文本是否相关的粒度较粗, 全局特征不会妨碍到预训练模型的计算结果. 而当判定相关的粒度变细时, 即在人工标注的数据集上, 缺少全局特征会降低对文档建模的有效性, 故而影响了文本相关度计算结果的准确性. 相对的, 本文提出的文本相关度计算方法通过构建篇章事件连通图从篇章级反映事件之间的联系, 故而能够抽取出与篇章级事件相关的特征, 其保证了选择的特征能够有效地反映出文本的核心事件, 故而获得了较好的文本相关度计算结果.

表 4 的结果与表 2 和 3 类似, 即本文提出的文本相关度计算方法在基于文本分类的标准数据集上取得了和基于文摘的方法类似的效果, 但是在人工标注的数据集上, 本文提出的方法获得了较好的

性能. 与表 2 不同的是, 表 4 中, 基于文摘的方法获得的文本相关度计算结果要优于有监督的方法且接近于预训练模型获得的结果. 获得上述结果的原因在于, 基于文摘的方法以挖掘到的文摘句为基础抽取特征来衡量文本之间的相关度. 文摘句能够体现文章的主题或是核心事件, 因此, 这种利用文摘句的方法能够较好地度量文本之间的相关度. 但是, 对于一篇文章来说, 其中的所有句子均应该为核心事件服务, 他们能够从不同侧面揭示文章的核心事件. 利用文摘句计算文本之间的相关度仅关注了从一个或几个句子中抽取的少量局部特征, 而忽略了文本全局信息对这些特征的影响. 故而基于文摘句的文本相关度计算方法的性能要低于本文提出的方法, 因为本文提出的方法中用于衡量文本相关度的特征是利用篇章事件连通图挖掘出来的, 这些特征考虑了文本的全局信息. 另外, 文摘句中也会存在一些与主题或核心事件无关的信息, 这些因素也会降低文本相关度计算方法的效果. 上述这些问题在相关度计算粒度较粗的标准数据集上不会显著的降低基于文摘的文本相关度计算方法的性能, 但是在粒度较细的人工标注数据集上就会显著的降低计算效果.

4 总结

篇章级事件表示及相关度计算有很多落地的应用场景, 比如对标题党的识别、描述同类事件的新闻聚类与推荐等. 本文即提出一种无监督的篇章级事件表示和相关度计算方法, 该方法无需大规模的标注数据, 减少了算法应用过程中的人力成本与时间成本. 本文所提方法通过构建篇章事件连通图以反映事件之间的联系, 进而通过计算图中节点的权值选择对篇章事件最具代表力的词语用以计算篇章级事件相关度. 具体而言, 本文所提方法的创新点可总结如下: (1) 在传统 TextRank 算法中引入了 EM 思想, 使得模型可以同时对句子和词语的重要性进行衡量; (2) 提出了以事件多边形为基本元素、事件触发词为核心节点的篇章事件连通图构建方式, 更加合理地揭示出文档所描述的核心事件. 最终的样例展示和统计实验结果均验证了本文所提方法的有效性.

参考文献

1 Sarawagi S. Information extraction. FNT Database, 2007, 1: 261–377

2 Riloff E. Automatically constructing a dictionary for information extraction tasks. In: Proceedings of the 11th National Conference on Artificial Intelligence, Washington, 1993. 811–816

3 Kim J, Moldovan D I. Acquisition of linguistic patterns for knowledge-based information extraction. IEEE Trans Knowl Data Eng, 1995, 7: 713–724

4 Chai J Y. Learning and generalization in the creation of information extraction systems. Dissertation for Ph.D. Degree. Durham: Duke University, 1998

5 Yangarber R. Scenario customization for information extraction. Dissertation for Ph.D. Degree. New York: New York University, 2001

6 Chen Y B, Liu S L, Zhang X, et al. Automatically labeled data generation for large scale event extraction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, 2017. 409–419

7 Turchi M, Zavarella V, Tanev H. Pattern learning for event extraction using monolingual statistical machine translation. In: Proceedings of Recent Advances in Natural Language Processing, 2011. 371–377

8 Li Q, Ji H, Huang L. Joint event extraction via structured prediction with global features. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, 2013. 73–82

9 Chieu H L, Ng H T. A maximum entropy approach to information extraction from semi-structured and free text. In: Proceedings of the 18th National Conference on Artificial Intelligence, Edmonton, 2002. 786–791

10 Alan R, Etzioni O, Clark S. Open domain event extraction from Twitter. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, 2012. 1104–1112

11 Ahn D. The stages of event extraction. In: Proceedings of Workshop on Annotations and Reasoning about Time and Events, Sydney, 2006
12 Ji H, Grishman R. Refining event extraction through unsupervised cross-document inference. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, Columbus, 2008. 254–262
13 Ji H. Cross-lingual predicate cluster acquisition to improve bilingual event extraction by inductive learning. In: Proceedings of NAACL HLT Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics, Boulder, 2009. 27–35
14 Liao S S, Grishman R. Using document level cross-event inference to improve event extraction. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, 2010. 789–797
15 Hong Y, Zhang J F, Ma B, et al. Using cross-entity inference to improve event extraction. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, 2011. 1127–1136
16 Liu S L, Liu K, He S Z, et al. A probabilistic soft logic based approach to exploiting latent and global information in event classification. In: Proceedings of the 13th AAAI Conference on Artificial Intelligence, Snowbird, 2016. 2993–2999
17 Chen Y B, Xu L H, Liu K, et al. Event extraction via dynamic multi-pooling convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, 2015. 167–176
18 Nguyen T H, Grishman R. Event detection and domain adaptation with convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, 2015. 365–371
19 Nguyen T H, Grishman R. Modeling skip-grams for event detection with convolutional neural networks. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, Austin, 2016. 886–891
20 Liu J, Chen Y B, Liu K, et al. Event detection via gated multilingual attention mechanism. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, 2018. 4865–4872
21 Yang S, Feng D W, Qiao L B, et al. Exploring pre-trained language models for event extraction and generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, 2019. 5284–5294
22 Liu X, Luo Z C, Huang H Y. Jointly multiple events extraction via attention-based graph information aggregation. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, Brussels, 2018. 1247–1256
23 Peng H R, Song Y Q, Roth D. Event detection and co-reference with minimal supervision. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, Austin, 2016. 392–402
24 Mihalcea R, Tarau P. TextRank: bringing order into text. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, Barcelona, 2004. 404–411
25 Lamberti F, Sanna A, Demartini C. A relation-based page rank algorithm for semantic web search engines. IEEE Trans Knowl Data Eng, 2009, 21: 123–136
26 Avrachenkov K, Litvak N, Nemirovsky D, et al. Monte carlo methods in pagerank computation: when one iteration is sufficient. SIAM J Numer Anal, 2007, 45: 890–904
27 Qiu L K, Zhang Y. ZORE: a syntax-based system for chinese open relation extraction. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, Doha, 2014. 1870–1880
28 Powers D M W. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. J Mach Learn Technol, 2011, 2: 37–63
29 Pennington J, Socher R, Manning C. Glove: global vectors for word representation. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, Doha, 2014. 1532–1543
30 Dongsuk O, Kwon S, Kim K, et al. Word sense disambiguation based on word similarity calculation using word vector representation from a knowledge-based graph. In: Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, 2018. 2704–2714
31 Che W X, Li Z H, Liu T. LTP: a Chinese language technology platform. In: Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, 2010. 13–16
32 Heafield K, Kayser M, Manning C D. Faster phrase-based decoding by refining feature state. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, 2014. 130–135
33 Huang F, Yates A. Open-domain semantic role labeling by modeling word spans. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, 2010. 968–978
34 Tan C Q, Wei F R, Wang W H, et al. Multiway attention networks for modeling sentence pair. In: Proceedings of the

27th International Joint Conference on Artificial Intelligence, Stockholm, 2018. 4411–4417
35 Gong Y C, Lu H, Zhang J. Natural language inference over interaction space. In: Proceedings of International Conference on Learning Representations, Vancouver, 2018
36 Kim S, Kang I, Kwak N. Semantic sentence matching with densely-connected recurrent and co-attentive information. In: Proceedings of AAAI Conference on Artificial Intelligence, Honolulu, 2019. 6586–6593
37 Devlin J, Chang M W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, 2019. 4171–4186
38 Cui Y M, Che W X, Liu T, et al. Pre-training with whole word masking for Chinese BERT. 2019. ArXiv:1906.08101
39 Zhang X X, Lapata M, Wei F R, et al. Neural latent extractive document summarization. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, Brussels, 2018. 779–784
40 Zhou Q Y, Yang N, Wei F R, et al. Neural document summarization by jointly learning to score and select sentences. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 2018. 654–663
41 Jadhav A, Rajan V. Extractive summarization with SWAP-NET: sentences and words from alternating pointer networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 2018. 142–151

Text correlation calculation based on passage-level event representation

Ming LIU[1,2], Zihao ZHENG[1], Bing QIN[1,2*], Yitong LIU[3] & Yang LI[3]

1. *School of Computer Science and Technology, Harbin Institute of Technology, Harbin* 150001*, China*;
2. *PENG CHENG Laboratory, Shenzhen* 518066*, China*;
3. *Tencent Technology* (*Beijing*) *Co., Ltd., Beijing* 100193*, China*
* Corresponding author. E-mail: qinb@ir.hit.edu.cn

Abstract Along with the explosion of web information, information flow service has attracted the attention of users. In this kind of service, how to measure the correlation between texts and further filter the redundant information collected from multiple sources becomes the key solution to meet the user's desire. Recently, the popular text correlation calculation methods mostly represent text as vector and then measure text similarity as text correlation. However, in information flow service, most of the texts are news, and the core element in a news is the event it stated. Therefore, we need a way to extract the core features that are related to the event stated by text, so we can accurately calculate text correlation via these extracted features. Unfortunately, recent event-related researches focus on the sentence-level. To calculate text correlation, we need to grasp the content of the text from the passage-level, which indicates that passage-level event analysis has more impact. To this end, we propose a passage-level event representation method based on sentence-level event extraction. It constructs a passage-level event connection graph based on the extracting results obtained from sentences. After that, it selects the important nodes in the graph as the representations of the passage-level events. Based on the passage-level representations, we can acquire text correlation. Experimental results indicate that our method outperforms conventional text correlation calculation methods.

Keywords passage event connection graph, passage-level event correlation, textrank, selection of key sentence, sentence-level connection graph

Ming LIU was born in 1981. He received his Ph.D. degree in computer application technology from Harbin Institute of Technology. Currently, he is an associate professor and Ph.D. supervisor at Harbin Institute of Technology. His research interests include data mining, knowledge graph, and machine reading.

Bing QIN was born in 1968. She received her Ph.D. degree in computer application technology from Harbin Institute of Technology. Currently, she is a professor and Ph.D. supervisor at Harbin Institute of Technology. Her research interests include data mining, sentiment analysis, and natural language processing.

Zihao ZHENG was born in 1998. Currently, he is an undergraduate at Harbin Institute of Technology. His research interests include knowledge graph, multimodal intelligence, and information extraction.

Yitong LIU was born in 1995. He received his M.A. degree in computer science and technology from Harbin Institute of Technology. Currently, he is a natural language processing algorithm engineer in Tencent, Beijing. His research interests include knowledge graph and data mining.