专业技术文本关键词抽取方法

宁祥东[1,2]，龚斌[1,2]，万林[1,2]，孙宇清[1,2,*]
1.山东大学，软件学院，济南，250101
2.教育部数字媒体技术工程研究中心，济南，250101
lcsxnxd@163.com，gb@sdu.edu.cn，wanlin@sdu.edu.cn，sun_yuqing@sdu.edu.cn

摘要

相关性和特异性对于专业技术文本关键词抽取问题至关重要，本文针对代码检索任务，综合语义信息、序列关系和句法结构提出了专业技术文本关键词抽取模型。采用预训练语言模型BERT提取文本抽象语义信息；采用序列关系和句法结构融合分析的方法构建语义关联图，以捕获词汇之间的长距离语义依赖关系；基于随机游走算法和词汇知识计算关键词权重，以兼顾关键词的相关性和特异性。在两个数据集和其他模型进行了性能比较，结果表明本模型抽取的关键词具有更好地相关性和特异性。

关键词： 关键词抽取 ； 句法结构 ； 语义信息 ； 专业文本

Keyword Extraction on Professional Technical Text

Xiangdong Ning [1,2], Bin Gong[1,2], Lin Wan[1,2], Yuqing Sun[1,2,*]
1.Shandong University,School of Software,Jinan,250101
2.Shandong University,ERC of Digital Media Technology, MoE,Jinan,250101
lcsxnxd@163.com，gb@sdu.edu.cn，wanlin@sdu.edu.cn，sun_yuqing@sdu.edu.cn

Abstract

For professional technical text keyword extraction problems, relevance and specificity are crucial, in order to achieve keyword extraction with relevance and specificity, we take semantic information, sequence relations and syntactic structure into account. Extraction of text semantic information using pre-trained language model BERT; We construct semantic association graph using sequence relation and syntactic structure, in order to capture long-distance semantic dependencies between words; We calculate keyword weights based on random walk algorithm and lexical knowledge, in order to take into account the relevance and specificity of keywords. Experimental results on professional text datasets show that keyword extracted by our model have better relevance and specificity.

Keywords: Keyword extraction , Syntactic structure , Semantic information , Profesional text

1 引言

开源代码平台为科研人员提供了分享和交流代码的环境，近几年，深度学习在自然语言处理、计算机视觉、生物计算等科研领域取得了很大成功。越来越多的深度学习模型和代码在开源平台分享，营造了可复用代码的生态环境，算法分享用户提供的代码描述文本包含功能和技术特征等专业词汇信息，以全球最大的开源代码存储库GitHub为例，其在2021年度报告中表明平台新增1600万个用户和6100万个新的代码库 (Liao et al., 2021)。专业文本的关键词抽取不仅要考虑查询关键词和代码描述文本的相关性，以提高代码检索的准确率，还要考虑关键词的特异性，以帮助用户探索新出现的代码。

基金项目：山东省自然科学基金(ZR2018ZB0420)

针对关键词抽取问题，现有方法主要包括三类：基于统计特征的文本关键词抽取方法主要依据词频、词长和词性等指标对候选关键词排序；基于图排序的关键词抽取方法将候选关键词视为节点，按照规则建立节点之间边，采用随机游走算法在词图上计算候选关键词的权重；基于主题模型的关键词抽取方法将候选关键词分配给文本包含的主题，选择每个主题下权重最大的词作为关键词。这些方法主要针对通用文本进行关键词抽取，主要关注关键词与文本的相关性，缺少考虑关键词的特异性，不适用于包含大量技术词汇的专业文本。

针对上述问题，本文主要贡献如下：

1）综合语义信息、序列关系和句法结构提出了专业技术文本关键词抽取模型。

2）采用预训练模型BERT作为文本编码器，提取文本抽象语义信息。

3）采用序列关系和句法结构融合分析的方法构建语义关联图，以捕获词汇之间的长距离语义依赖关系。

4）基于随机游走算法和词汇知识计算关键词权重，以兼顾关键词的相关性和特异性。

在两个数据集和其他模型进行了性能比较，结果表明本模型抽取的关键词具有更好地相关性；在关键词特异性分析中，基于随机游走算法和词汇知识的方法更好地提升了关键词的特异性；通过析构分析，验证了依存句法知识对模型性能带来的收益最大。

论文其他内容组织如下：相关工作中介绍了主流的关键词抽取方法，并分析了现有方法的优势和不足；对专业技术文本关键词抽取模型进行了细节介绍；在两个包含专业词汇的数据集上与现有方法进行了性能对比分析，讨论了本文方法的各部分对模型性能的影响，分析了关键词的重要性和特异性。

2 相关工作

针对关键词抽取问题，最具代表性的是基于图排序的关键词抽取方法，该方法主要思想是将描述文本中的候选关键词视为节点，然后按照一定的规则建立节点之间的边，采用随机游走算法 (Blanco et al., 2012)在词图上计算词汇权重。例如，Mihalcea (2004)等人提出了基于图排序的TextRank算法，使用文本中的单词作为节点，依据共现词汇构建边。目前已经提出了多种基于TextRank的方法，例如Wan (2008)等人提出的SingleRank方法将滑动窗口中单词的共现次数分配给词图中边的权重，该方法只能使用单个文本的信息来构建图。为了更好地表示图中节点之间的关系，越来越多的工作倾向于使用词之间的语义关系来计算图中边的权重。Tsatsaronis (2010)等人提出了SemanticRank的方法，该方法利用语义关系从文档中提取关键词和句子，在实验中证明该方法优于TextRank方法。一些工作将先验知识添加到图中的节点以强调单词的重要性，例如单词的位置、TFIDF值等。为了进一步提高TextRank算法的关键词抽取效果，Florescu (2017)等人提出了PositionRank算法，这是一种从学术文档中抽取关键词的无监督方法，该方法在词权值迭代的时候融入位置信息。Caragea (2014)等人比较了基于图的关键词抽取方法的各种中心性度量，结果表明，简单的中心性度量的结果与TextRank方法一样。这类方法能够融入深层次的文本语义信息和句法知识，但是受到分词结果的影响较大。

另一类代表性的方法是基于统计特征的关键词抽取方法和基于主题模型的关键词抽取方法。基于统计特征的方法依据统计指标对候选关键词排序 (Wang et al., 2020; Campos et al., 2020)，统计指标指标通常包括词频、词长和词性等，例如将候选关键词的TFIDF (Wang et al., 2020)值作为统计特征，依据TFIDF值的大小抽取出关键词集合，但是这类统计特征忽略了单词自身的属性，因此Campos (2020)等人提出了使用单词词性、在描述文本中出现的位置等指标为候选关键词设置不同的权重。这类方法运行速度快，但是不能提取深层次的文本语义信息。基于主题模型的方法一般是将候选关键词分配给文本包含的主题，选择每个主题下权重最大的词汇作为关键词，例如，Bougouin (2013)等人提出了一种基于主题的TopicRank方法，该方法将文档表示为一个完整的图，其中顶点不是单词而是主题。这类方法能够分析文档中的潜在主题，但是，不适用于频繁出现的关键词，难以适用于专业文本的关键词抽取。

部分工作将关键词抽取问题视为文本分类问题，将关键词库中的关键词作为类别标签，即对描述文本进行多标签分类。例如，基于循环神经网络的方法将文本中的词向量逐个输入到神经网络单元中，使用隐含层的最后一个输出来预测文本的标签 (Zheng et al., 2019)。基于卷积神经网络的方法将词向量拼接成矩阵，然后将其输入卷积神经网络后得到的文本向量，将文本向量输入到分类函数中来预测类别标签 (Wang et al., 2021; Jacovi et al., 2018)。以上两种方法由于隐藏数据的不可读性，导致了可解释性较差 (Sun et al., 2019)。Yang (2016)等人提出了包

括两个编码器和两个注意力层的HAN方法，该模型先将输入词汇聚合成句子向量，然后基于句子向量聚合成文本向量，通过注意力机制可以分析词和句子对类别的权重影响。这类方法一般需要依赖于大型训练数据集。

3 专业技术文本关键词抽取模型

3.1 问题描述

在基于关键词代码检索平台中，为了提高代码检索的准确率和帮助用户探索新出现的代码，专业技术文本的关键词抽取应满足以下三个性质：

- **相关性**：抽取出的技术特征关键词能够代表代码使用的技术和实现的功能。
- **重要性**：针对抽取的有限个关键词，要求按照重要程度排序。
- **特异性**：抽取出的技术特征关键词相对于代码检索平台中其他技术特征关键词的显著程度，有助于帮助用户探索新出现的代码。

3.2 整体框架

针对专业文本的关键词抽取问题，本文综合语义信息、序列关系和句法结构提出了专业技术文本关键词抽取模型。如图1所示，首先对代码描述文本进行删除停用词、保留相关词性的词和删除无意义的标点符号等预处理；为了提取文本的抽象语义信息，采用预训练语言模型BERT作为文本编码器，进而得到候选关键词向量列表；采用序列关系和句法结构融合分析的方法构建语义关联图，以捕获词汇之间的长距离语义依赖关系，图中的节点表示候选关键词，边的权重为候选关键词向量的余弦相似度值；基于随机游走算法和词汇知识计算关键词分数，以兼顾关键词的相关性和特异性；依据候选关键词的分数进行倒排序，使用语言模型得到TOP-K个关键词。第 3.3节至 3.5节对模型进行细节介绍。

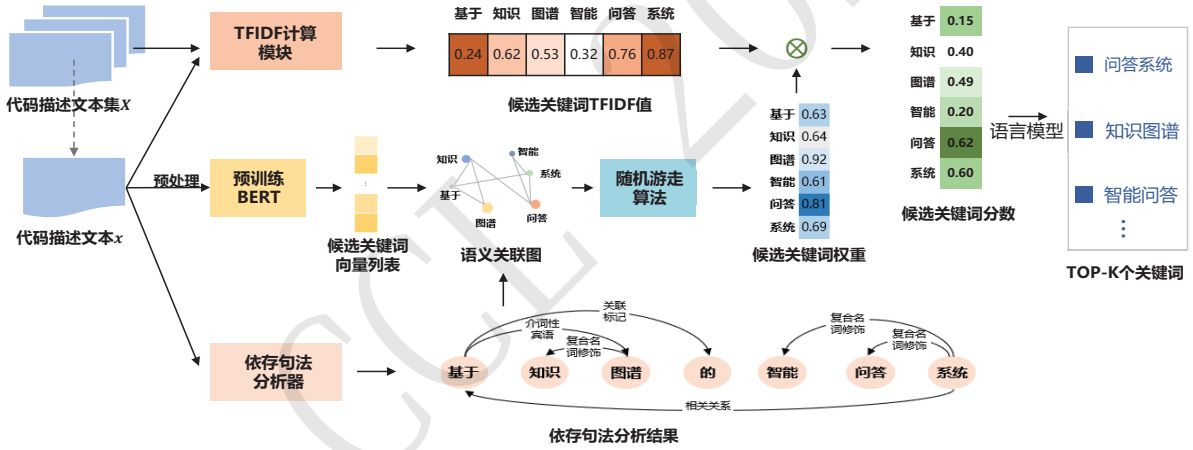


图 1. 专业技术文本关键词抽取模型

3.3 基于预训练语言模型的词汇编码

为了抽取出的关键词与文本更相关，我们对代码描述文本x进行删除停用词、保留相关词性的词和删除标点符号等预处理，经过预处理后得到候选关键词集合V_x。为了提取文本的抽象语义信息，本文采用预训练语言模型BERT (Devlin et al., 2019)对词汇上下文进行语义编码，依据编码结果得到候选关键词向量。

$$V_x = \{v_1, v_2, ..., v_n\} \tag{1}$$

$$[\boldsymbol{e}_{v_1}, \boldsymbol{e}_{v_2} ... \boldsymbol{e}_{v_n}] = BERT\,(v_1, v_2 ... v_n) \tag{2}$$

其中，v_i表示第i个候选关键词，n表示候选关键词数量，V_x表示候选关键词集合，$\boldsymbol{e}_{v_i}$表示第i个候选关键词向量。

3.4 融合序列关系和句法结构的语义关联图构建

本文基于共现词汇得到序列关系，融合序列关系和句法结构 (Chen et al., 2014)来构建建语义关联图的边E_x，语义关联图中的节点v_i表示候选关键词，边的权重为候选关键词向量的余弦相似度W_x，语义关联图是一个无向加权图。

$$E_x = \{(v_i, v_j) \,|v_i \in V_x, v_j \in V_x\} \tag{3}$$

$$w_{ij} = \begin{cases} cos\left(\boldsymbol{e}_{v_i}, \boldsymbol{e}_{v_j}\right), (v_i, v_j) \in E_x \\ 0, \text{其他} \end{cases} \tag{4}$$

$$W_x = \{w_{ij} | 1 \leq i \leq n, 1 \leq j \leq n\} \tag{5}$$

$$G_x = (V_x, E_x, W_x) \tag{6}$$

其中，E_x表示候选关键词存在的边集合，W_x表示边权重集合，w_{ij}表示v_i和v_j词向量的余弦相似度，G_x是语义关联图。

3.5 基于随机游走算法和词汇知识的关键词权重计算

本文模型综合考虑了关键词的相关性和特异性，采用随机游走算法 (Blanco et al., 2012)在语义关联图G_x上进行迭代计算后得到每个候选关键词的权重$WS_x\left(v_i\right)$，使得抽取出的关键词能够与文本更相关，具体计算公式如下所示。

$$WS_x\left(v_i\right) = (1-d) + d \times \sum_{v_j \in Nei(v_i)} \frac{w_{ij}}{\sum_{v_k \in Nei\left(v_j\right)} w_{jk}} WS_x\left(v_j\right) \tag{7}$$

其中，$WS_x\left(v_i\right)$为候选关键词v_i的权重，$WS_x\left(v_j\right)$表示上一次迭代后节点v_j的权重，$Nei(v)$表示v的邻节点集合，d为阻尼系数。

为了更好的解释基于随机游走算法计算候选关键词权重的过程，在此对计算过程进行详细说明：计算候选关键词权重的过程是一个马尔可夫过程，根据词向量的余弦相似度值可以得到词汇相似度矩阵$S_{n\times n}$，矩阵$S_{n\times n}$是一个对称矩阵，并且对角线上的元素全部取0，设定所有候选关键词的初始权重$\boldsymbol{B}_0$为该候选关键词的$tfidf$值，具体计算公式如下：

$$S_{n\times n} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \tag{8}$$

$$\boldsymbol{B}_i = S_{n\times n} \boldsymbol{B}_{i-1} \tag{9}$$

$$\boldsymbol{B}_0 = \left[tfidf_x\left(v_1\right), tfidf_x\left(v_2\right) ... tfidf_x\left(v_n\right)\right]^T \tag{10}$$

其中，$S_{n\times n}$表示候选关键词相似度矩阵，$\boldsymbol{B}_0$中的元素为所有候选关键词的初始值，$\boldsymbol{B}_i$表示第i轮计算后候选关键词的权重，$tfidf_x\left(v_i\right)$表示第i个候选关键词的$tfidf$值，只有当$\boldsymbol{B}_i$与$\boldsymbol{B}_{i-1}$的差值非常小且接近于零时达到算法收敛，算法收敛后可以得到候选词的权重。

$$tf_x\left(v_i\right) = \frac{count\left(v_i, x\right)}{size\left(x\right)} \tag{11}$$

其中，x表示代码描述文本，$size\left(x\right)$表示代码描述文本x中包含的候选关键词个数，$count\left(v_i, x\right)$表示代码描述文本x中包含第i个候选关键词的个数。

$$idf_x\left(v_i\right) = log\left(\frac{size(X)}{count(v_i, X)+1}\right) \tag{12}$$

第二十一届中国计算语言学大会论文集，第143页-第154页，南昌，中国，2022年10月14日至16日。

其中，X表示代码描述文本集，$size(X)$表示代码描述文本集中包含的代码描述文本数量，$count(v_i, X)$表示包含第i个候选关键词的代码描述文本的数量。

$$tfidf_x(v_i) = tf_x(v_i) \times idf_x(v_i) \tag{13}$$

tf_x表示第i个候选关键词v_i在代码描述文本x中的词频，idf_x表示第i个候选词v_i在整个代码描述文本集合X中的逆向文本频率。

本文模型采用词汇的$tfidf$值作为词汇知识，以兼顾关键词的相关性和特异性，将公式7得到的权重$WS_x(v_i)$与词汇知识进行融合，得到候选关键词分数$Score(v_i)$。为了更准确的抽取代码描述文本中的专业词汇，本文依据GitHub平台提供的代码主题，创建了一个专业词汇列表，如果候选关键词是专业词汇，那么该候选关键词的权重相对于其他候选关键词被设置为一个最大值，候选词分数的计算如公式14所示。

$$Score(v_i) = WS_x(v_i) \times tfidf_x(v_i) \tag{14}$$

$Score(v_i)$表示第i个候选词的分数，依据分数对词汇进行倒排序，使用语言模型 (Pauls et al., 2011)得到TOP-K个关键词作为技术特征关键词。

4 实验与结果分析

4.1 数据集

针对专业文本的关键词抽取问题，我们在实验中选择了两个公开且包含专业词汇的KDD、WWW数据集来验证模型的有效性。两个数据集均为ACM会议和万维网会议的研究论文，由Li (2021)等人的论文提供。

实验数据集的统计信息如表1所示，两个数据集均由论文摘要和关键词组成，数据集中的关键词均由论文作者给出，所以将作者给出的关键词视为参考关键词，将论文摘要视为描述文本。本文只保留至少包含两个句子和一个关键词的文档，KDD和WWW数据集分别包含704和1248个文档。分别在两个数据集上进行模型性能分析、关键词重要性分析、关键词特异性分析和模型析构分析。

数据集	文档总数	文档平均长度	文档平均关键词个数	关键词在文中存在比
KDD	704	204	4.16	68.12
WWW	1248	174	4.78	64.97

表 1. 实验数据集统计信息表

4.2 评价指标

（1）基于统计的精准率和召回率

本文使用精准率和召回率衡量关键词抽取算法的准确程度，命中集合指算法抽取出的关键词集合与参考关键词集合的交集，精准率$Precision$表示关键词抽取模型的准确程度，是命中集合与算法抽取出的关键词集合大小的比值。召回率$Recall$表示模型抽取的关键词对文本的覆盖程度，是命中集合与参考关键词集合大小的比值。为了避免精准率和召回率指标的冲突，我们使用精确率和召回率的调和平均数F_1分数来评估模型性能，公式如15所示：

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{15}$$

（2）基于语义关系的精准率和召回率

基于统计的评价指标只能评估精准匹配的关键词，不能反映抽取关键词与参考关键词之间的语义关系，例如同义词。为此，我们设计了基于语义关系的评价指标。通过预训练语言模型BERT对词汇上下文进行编码得到词向量，通过计算参考关键词和抽取关键词向量的内积，得到语义相似性矩阵。语义精准率和召回率为参考关键词和抽取关键词最大相似性得分的累加，然后归一化，计算公式如16-18：

$$Precision_s = \frac{1}{|\hat{Y}|}\sum_{\hat{y}_i \in \hat{Y}} max_{y_j \in Y}\left(\hat{\boldsymbol{w}}_i{}^T \cdot \boldsymbol{w}_j\right) \tag{16}$$

$$Recall_s = \frac{1}{|Y|}\sum_{y_j \in Y} max_{\hat{y}_i \in \hat{Y}}\left(\hat{\boldsymbol{w}}_i{}^T \cdot \boldsymbol{w}_j\right) \tag{17}$$

$$F_1^s = 2 \times \frac{Precision_s \times Recall_s}{Precision_s + Recall_s} \tag{18}$$

其中，$\boldsymbol{w}_i$表示算法抽取出第i个关键词的词向量。$\boldsymbol{w}_j$表示第j个参考关键词的词向量，y_j表示第j个参考关键词，$\hat{y}_i$表示算法抽取出的第i个关键词。

（3）排名倒数指标

本文考虑了抽取关键词的排列顺序，使用排名倒数评估关键词排列的重要程度，排名倒数表示所有参考关键词在算法抽取的关键词集合中位置倒数的期望。如果参考关键词在抽取出的关键词集合中的位置越靠前，MRR值就会越大，公式如19：

$$MRR = \frac{1}{|Y|}\sum_{j=1}^{|Y|}\left(\frac{1}{Rank_{y_j}}\right) \tag{19}$$

其中，$Rank_{y_j}$表示参考关键词集合中第j个关键词在抽取的关键词序列中的序号。

（4）特异性指标

特异性表示抽取出关键词的显著程度，IDF值适用于评估算法抽取关键词的显著程度，TF值适用于评估关键词与代码描述文本的相关程度，为了兼顾关键词的显著程度和相关程度，本文使用TF和IDF的调和平均数$Specific$来评估关键词的特异性，公式如20-22：

$$IDF = \frac{1}{|\hat{Y}|}\sum_{i=1}^{|\hat{Y}|} log\frac{size(X)}{count(\hat{y}_i, X)} \tag{20}$$

$$TF = \frac{1}{|\hat{Y}|}\sum_{i=1}^{|\hat{Y}|}\frac{count(\hat{y}_i, x)}{size(x)} \tag{21}$$

$$Specific = 2 \times \frac{TF \times IDF}{TF + IDF} \tag{22}$$

其中，代码描述文本集中包含代码描述文本的数量为$size(X)$，包含算法抽取出的第i个关键词的文档数量为$count(\hat{y}_i, X)$。代码描述文本中词的数量为$size(x)$，算法抽取出的第i个关键词在代码描述文本x中出现的次数为$count(\hat{y}_i, x)$。

4.3 对比方法

对比模型如下：

- **TripleRank** (Li et al., 2021)：提出了一个无监督的关键词抽取TripleRank方法，该方法考虑了关键词位置、语义多样性和覆盖率的特征，依据这三种特征计算候选关键词的得分。
- **ISKE** (Chi et al., 2021)：提出了一种不依赖于外部资源的关键词抽取算法。使用迭代句子对单词进行排名，依据句子的语义信息生成候选关键词列表。使用加权信息初始化词的值，并使用这些值生成句子分数，依据句子的分数来更新候选关键词的值。
- **GTCRank** (Li et al., 2019)：提出了一种使用基于图排序和基于主题聚类的方法来提取关键词的无监督算法，使用基于图排序的方法来描述两个词之间的相关性，并使用基于主题聚类的方法将语义信息嵌入到词中。

- **PositionRank** (Florescu et al., 2017)：提出了一个用于从学术文档中抽取关键词的无监督算法，该方法在迭代计算词权重的过程中融入了位置信息，融入方式有两种，一种是融入了该词出现的所有位置，另外一种是融入了该词出现的第一个位置。

- **YAKE** (Campos et al., 2020)：提出了一个无监督的关键词提取的YAKE算法，该算法依据从单个文档中提取的统计特征来选择文本中重要的关键词，统计特征主要包括候选关键词位置、词频等。

- **TPR** (Liu et al., 2010)：提出了将在单个词图上随机游走分解为在多个不同主题上随机游走的算法，在不同主题下分别计算候选关键词的权重，最后依据文档的主题分布来计算单词的最终排名分数。

- **RSKeyRank**：本文设计的专业技术文本关键词抽取模型，记为RSKeyRank算法，其输入是代码描述文本x，输出为RSKeyRank算法从代码描述文本中抽取到的关键词集合$\hat{Y}$，参考关键词集合为Y。

- **RSKeyRank-TFIDF**：表示RSKeyRank算法没有使用TFIDF计算模块。

- **RSKeyRank-BERT**：表示RSKeyRank算法没有使用预训练语言模型BERT。

- **RSKeyRank-Syntax**：表示RSKeyRank算法没有融入句法知识。

- **RSKeyRank-Syntax-BERT**：表示RSKeyRank算法既没有使用预训练语言模型BERT，也没有融入句法知识。此时，仅基于序列关系构建语义关联图中的边，边的权重为1。

4.4 模型性能分析

（1）基于统计的精准率和召回率

为了分析本文模型性能，我们使用精准率、召回率和F_1分数在KDD和WWW数据集上与上述模型进行了性能对比，如表2所示，表格上半部分是在KDD数据集上关键词抽取的效果，下半部分是在WWW数据集上关键词抽取的效果，TOP5和TOP10分别表示算法抽取5个和10个关键词。

从整体上来看，本文方法在两个数据集上都取得了最好的结果，说明本文方法抽取的关键词与文本更相关。在KDD数据集上抽取5个和10个关键词时，F_1分数分别达到了14.7%和15.6%，在WWW数据集上抽取5个和10个关键词时，F_1分数分别达到了16.5%和16.7%。在KDD数据集上抽取5个关键词时，RSKeyRank算法相较于YAKE算法和TripleRank算法的F_1分数提升了11.3%和2.2%，这表明采用图排序的关键词抽取算法相较于基于统计的关键词抽取算法有较大提升。RSKeyRank算法相较于ISKE算法的F_1分数提升了2.4%，这表明本文方法相较于使用迭代句子对单词进行排序的算法在关键词抽取任务上有较大提升。RSKeyRank算法相较于基于图排序的PositionRank、GTCRank和TPR算法的F_1分数分别提升了2.5%、5.3%和6.2%，这表明融入预训练语言模型BERT和句法知识可以提升关键词抽取模型的性能。

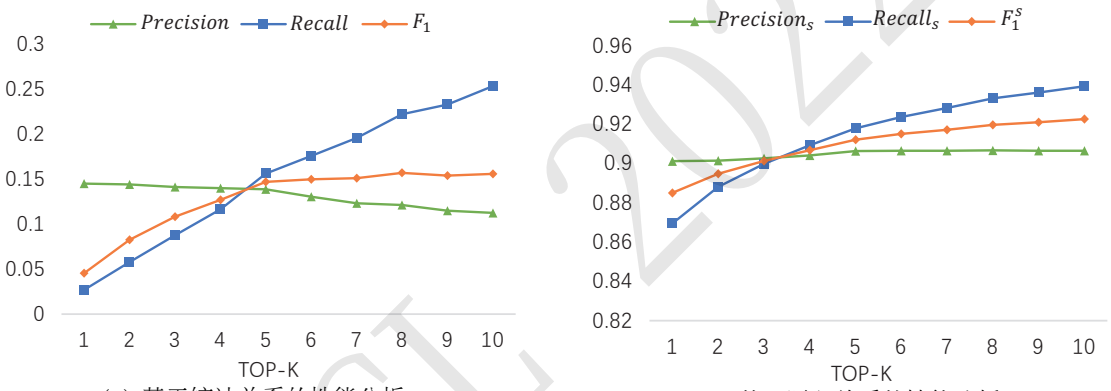
如图2(a)所示，横坐标表示RSKeyRank算法抽取关键词的个数，在KDD和WWW数据集上随着抽取关键词个数的增多，精准率一直在下降，召回率一直在上升，F_1分数则是先上升后保持不变。召回率是命中集合与参考关键词集合大小的比值，图2(a)中召回率一直在上升，表明随着RSKeyRank算法抽取出关键词个数的增多，抽取出关键词的覆盖程度就越高。图2(a)中精准率一直在下降，表明随着RSKeyRank算法抽取出关键词个数的增多，关键词抽取模型的准确程度在下降。F_1分数是精准率和召回率的调和平均，在图2(a)中F_1分数先上升后保持不变，RSKeyRank算法抽取出关键词的个数介于1和5之间时，召回率的上升速度比精准率的下降速度快，抽取出的关键词个数介于6和10之间时，精准率的下降速度与召回率的上升速度基本一致，这表明本文算法抽取的关键词个数大于等于5个时模型的性能趋于稳定。

（2）基于语义关系的精准率和召回率

基于统计的评价指标只能评估精准匹配的关键词，不能反映抽取关键词与参考关键词之间的语义关系，例如同义词。为此，我们使用基于语义关系的精准率和召回

数据集	方法	TOP5			TOP10		
		Precision%	*Recall*%	F_1%	*Precision*%	*Recall*%	F_1%
KDD	TripleRank	11.9	14.5	12.5	9.2	19.8	11.8
	ISKE	12.0	14.3	12.3	9.1	22.0	12.2
	GTCRank	8.7	11.1	9.4	7.9	20.1	11.2
	PositionRank	11.7	14.2	12.2	9.0	19.9	11.7
	YAKE	3.1	4.0	3.4	3.5	8.8	4.8
	TPR	8.1	9.7	8.5	7.4	16.4	9.7
	RSKeyRank	**13.9**	**15.6**	**14.7**	**11.3**	**25.3**	**15.6**
WWW	TripleRank	12.9	14.2	12.9	10.1	19.6	12.5
	ISKE	12.8	13.9	12.7	10.2	19.8	12.6
	GTCRank	9.9	11.3	10.1	8.8	19.4	11.6
	PositionRank	12.4	13.6	12.3	9.9	19.7	12.3
	YAKE	4.4	5.0	4.5	3.9	8.6	5.1
	TPR	9.4	10.2	9.3	8.5	16.7	10.5
	RSKeyRank	**15.5**	**17.6**	**16.5**	**12.2**	**26.7**	**16.7**

表 2. 关键词抽取模型的性能对比



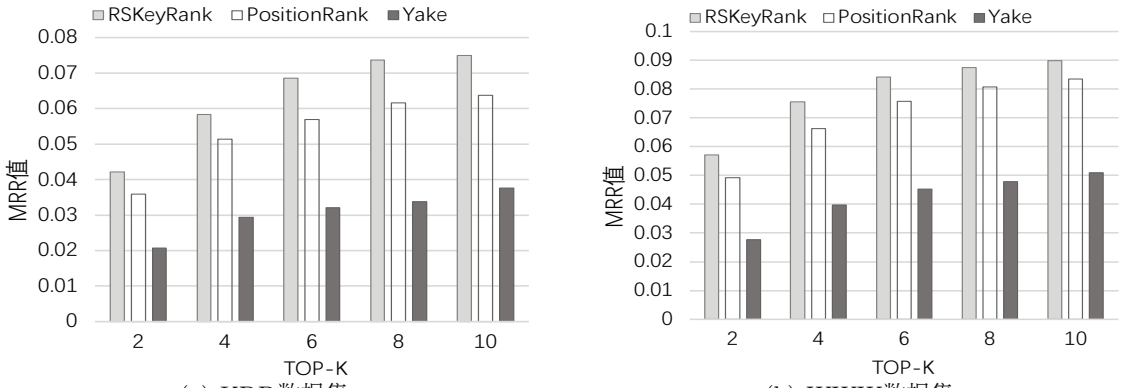
(a) 基于统计关系的性能分析　　(b) 基于语义关系的性能分析

图 2. KDD数据集上模型性能分析

率作为基于统计评价指标的补充。如图2(b)所示，横坐标表示RSKeyRank算法抽取的关键词个数，在KDD和WWW数据集上随着抽取的关键词个数的增多，$Precision_s$保持不变，$Recall_s$和F_1^s分数一直在上升，这表明随着RSKeyRank算法抽取出关键词个数的增多，算法抽取的关键词与参考关键词的语义关系就越强。图2(b)中精准率保持不变，因为RSKeyRank是无监督算法，不具备自学习的能力。

4.5 关键词重要性分析

为了评估指抽取出关键词排列顺序的重要程度，本文采用排名倒数评价关键词的重要性。如图3所示，横坐标表示各个模型抽取关键词的个数，纵坐标表示MRR值。只有参考关键词存在于抽取关键词集合时，参考关键词才有排名倒数，所以关键词的重要性和算法的性能保持一致。RSKeyRank算法的性能比Yake和PositionRank算法的性能要高，所以RSKeyRank算法相较于Yake和PositionRank算法的MRR值均达到了最高，说明了本文方法抽取的关键词排列顺序相较于其他方法更为合适。随着RSKeyRank算法抽取的关键词个数增多时，算法抽取的关键词与参考关键词相匹配的个数就越多，在两个数据集上MRR值一直在上升。



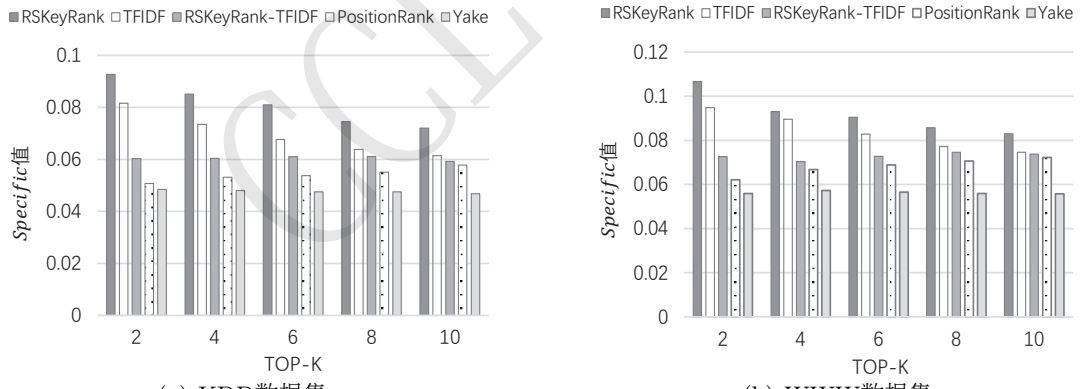
(a) KDD数据集　　(b) WWW数据集

图 3. 关键词的重要性分析

4.6 关键词特异性分析

为了评估算法抽取出关键词的显著程度，本文采用$Specific$评价指标评估算法抽取出关键词的特异性。如图4所示，横坐标表示各个模型抽取的关键词个数，纵坐标表示$Specific$值。

本文设计的算法相较于其他四种算法的$Specific$值均达到了最高，这表明RSKeyRank算法抽取到的关键词既与代码描述文本相关，又具有特异性。RSKeyRank算法相较于TFIDF方法在特异性评价指标的提升最小，因为RSKeyRank算法和TFIDF方法均使用了IDF计算模块。如果本文设计的方法没有使用TFIDF计算模块，那么RSKeyRank算法抽取出关键词的特异性值有较大的下降，但是会高于基于图排序的PositionRank算法和基于统计方法的Yake算法，因为根据 4.4节的性能分析，得知本文设计的算法抽取出的关键词与代码描述文本的相关性最高，所以特异性值高于PositionRank算法和Yake算法。这表明随机游走算法和词汇知识的融合更好地提升了关键词的特异性。RSKeyRank算法和TFIDF方法抽取出的关键词的特异性值明显高于PositionRank和Yake算法，因为PositionRank和Yake算法仅考虑了关键词与代码描述文本的相关性。PositionRank方法比Yake方法抽取出关键词的特异性值高，因为PositionRank算法抽取的关键词与代码描述文本的相关性高于Yake算法。



(a) KDD数据集　　(b) WWW数据集

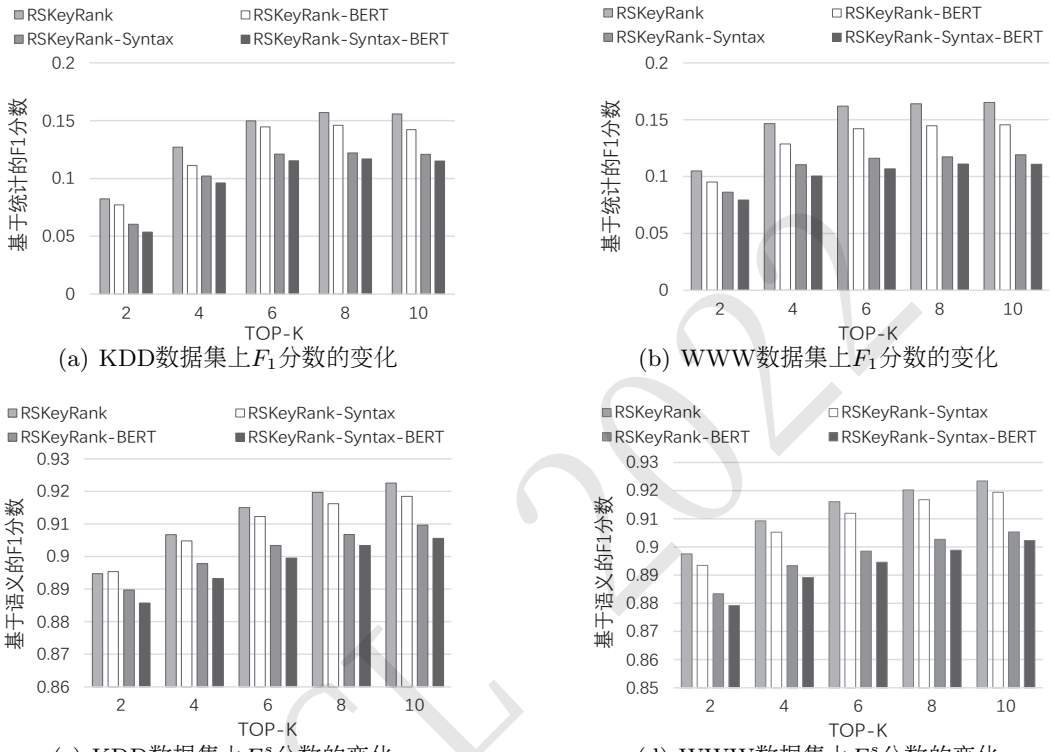
图 4. 关键词的特异性分析

4.7 模型析构分析

为了验证RSKeyRank算法各部分结构对模型带来的收益，本文对句法知识模块、预训练语言模型BERT进行有效拆分，通过基于统计和基于语义关系的评价指标在两个数据集上验证不同模块对RSKeyRank算法带来的收益，结果如图5所示。图5(a)和图5(b)表示各个模型在两个数据集上基于统计的F_1分数的变化，图5(c)和图5(d)表示各个模型在两个数据集上基于语义关系的F_1^s分数的变化。横坐标表示RSKeyRank算法抽取关键词的个数，纵坐标分别表示F_1分数

和F_1^s分数。

图5(a)和图5(b)中RSKeyRank相比于其他三个模型的F_1分数都要高，其他三个模型与RSKeyRank之间F_1分数的差距表示对模型性能带来的收益。这表明融入了句法知识对模型性能带来的收益最大，使用预训练语言模型BERT对模型性能带来的收益次之。本文设计的RSKeyRank算法是基于图排序的方法，图中的节点表示候选关键词，节点之间的边表示存在关系的两个关键词，因为句法知识可以捕获词汇的长距离语义依赖关系，使得词图更接近于真实分布，所以句法知识对模型带来了收益。图5(c)和图5(d)中RSKeyRank相较于其他三个模型的F_1^s分数最高,其他三个模型与RSKeyRank之间F_1^s分数的差距表示对模型带来语义相关性的收益，这表明使用预训练语言模型BERT对模型抽取出关键词与参考关键词之间的语义相关性带来的收益最大，融入句法知识对关键词的语义相关性带来的收益次之，因为预训练语言模型BERT可以捕获词汇之间的语义关系。



(a) KDD数据集上F_1分数的变化　(b) WWW数据集上F_1分数的变化

(c) KDD数据集上F_1^s分数的变化　(d) WWW数据集上F_1^s分数的变化

图 5. 模型析构分析实验结果

5 总结

针对专业技术文本的关键词抽取问题，本文综合语义信息、序列关系和句法结构提出了专业技术文本关键词抽取模型。采用预训练模型BERT作为文本编码器，提取文本抽象语义信息；采用序列关系和句法结构融合分析的方法构建语义关联图，以捕获词汇之间的长距离语义依赖关系；基于随机游走算法和词汇知识计算关键词权重，以兼顾关键词的相关性和特异性。在两个数据集和其他模型进行了性能比较，结果表明本模型抽取的关键词具有更好地相关性；在关键词特异性分析中，基于随机游走算法和词汇知识的方法更好地提升了关键词的特异性；通过析构分析，验证了依存句法知识对模型性能带来的收益最大。在今后工作中，计划进一步融合代码描述文本和代码结构进行关键词抽取，以提高模型性能。

参考文献

Adam Pauls, Dan Klein. 2011. Faster and Amaller N-gram language models. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011: 258-267.

Adrien Bougouin, Florian Boudin and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction. *International Joint Conference on Natural Language Processing*. 2013: 543-551.

Alon Jacovi, Oren Sar Shalom, Yoav Goldberg. 2018. Understanding convolutional neural networks for text classification. *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 2018: 56-65.

Ammar Ismael Kadhim. 2019. Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, 2019, 52(1): 273-292.

Corina Florescu and Cornelia Caragea. 2017. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 2017: 1105-1115.

Cornelia Caragea, Florin Bulgarov, Andreea Godea et al. 2014. Citation-enhanced keyphrase extraction from research papers: A supervised approach. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 2014: 1435-1446.

Danqi Chen, Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 2014: 740-750.

George Tsatsaronis, Iraklis Varlamis and Kjetil Nørvåg. 2010. SemanticRank: ranking keywords and sentences using semantic graphs. *Proceedings of the 23rd International Conference on Computational Linguistics*. 2010: 1074-1082.

Haitao Wang, Keke Tian, Zhengjiang Wu et al. 2021. A short text classification method based on convolutional neural network and semantic extension. *International Journal of Computational Intelligence Systems*, 2021, 14(1): 367-375.

Jin Zheng, Limin Zheng. 2019. A hybrid bidirectional recurrent convolutional neural network attention-based model for text classification. *IEEE Access*, 2019, 7: 106673-106685.

Jacob Devlin, Ming-Wei Chang, Kenton Lee et al. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019: 4171-4186.

Ling Chi and Liang Hu. 2021. ISKE: An unsupervised automatic keyphrase extraction approach using the iterated sentences based on graph method. *Knowledge-Based Systems*, 2021, 223: 107014.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali et al. 2020. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 2020, 509: 257-289.

Roi Blanco, Christina Lioma. 2012. Graph-based term weighting for information retrieval. *Information Retrieval*, 2012, 15(1): 54-92.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. 2004: 404-411.

Shengli Sun, Qingfeng Sun, Kevin Zhou et al. 2019. Hierarchical attention prototypical networks for few-shot text classification. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 2019: 476-485.

Tuohang Li, Liang Hu, Hongtu Li et al. 2021. TripleRank: An unsupervised keyphrase extraction algorithm. *Knowledge-Based Systems*, 2021, 219:106846.

Tengfei Li, Liang Hu, Jianfeng Chu et al. 2019. An Unsupervised approach for keyphrase extraction using within-collection resources. *IEEE Access*, 2019, 7: 126088-126097.

Xinyun Wang and Hongyun Ning. 2020. TFIDF Keyword extraction method combining context and semantic classification. *Proceedings of the 3rd International Conference on Data Science and Information Technology*. 2020: 123-128.

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*. 2008: 855-860.

Xiangke Mao, Shaobin Huang, Rongsheng Li et al. 2020. Automatic keywords extraction based on co-occurrence and semantic relationships between words. *IEEE Access*, 2020, 8: 117528-117538.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng et al. 2010. Automatic keyphrase extraction via topic decomposition. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. 2010: 366-376.

Zhifang Liao, Yiqi Zhao and Shengzong Liu. 2021. The measurement of the software ecosystem's productivity with github. *Computer Systems Science and Engineering*, 36(1): 239-258.

Zichao Yang, Diyi Yang, Chris Dyer et al. 2016. A fusion model-based label embedding and self-interaction attention for text classification. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016: 1480-1489.