基于分类的term重要性识别方法

邱云飞[1] 鲍 莉[1] 邵良杉[2]
(辽宁工程技术大学软件学院 葫芦岛 125100)[1] (辽宁工程技术大学系统工程研究所 阜新 123000)[2]

摘 要 在传统的搜索引擎和信息检索中,用户Query中的term-weight通常是以一种上下文无关的方式得到的。现有的大多数信息检索技术都使用词袋方法,例如布尔模型、向量空间模型和概率模型等,这些方法均没有考虑Query中term之间的相关性。为了能够充分利用Query中的信息来提高term-weight的准确度,提出了一种有监督的机器学习方法来学习用户Query中的term-weight。该方法基于分类的方法,并引入了句法分析作为分类的一项重要的特征来训练模型。考虑用户Query中term之间的关系后,既避免了由Query到单个term的信息丢失,又增加了短文本的特征,同时使分类器实现软输出,能够给term的重要程度一个更为准确的量化值。

关键词 分类,依存句法分析,查询词权重,查询分析,term重要性,搜索引擎,信息检索
中图法分类号 TP311.1 **文献标识码** A

Term Importance Identification Method Based on Classification

QIU Yun-fei[1] BAO Li[1] SHAO Liang-shan[2]
(School of Software, Liaoning Technical University, Huludao 125100, China)[1]
(System Engineering Institute, Liaoning Technical University, Fuxin 123000, China)[2]

Abstract In the field of traditional search engines and information retrieval, term weights for the input query are typically derived in a context independent fashion. Most information retrieval techniques employ bag-of-words approaches like Boolean models, vector-space models and other probabilistic ranking approaches to obtain term-weight of a term in a query. However, all these algorithms treat terms independently, and do not take the relationship among the terms. This paper employed supervised machine learning based on classification and syntactic parsing to derive a context-sensitive and query-dependent term weight for each word in a search query. By taking the result of syntactic parsing as a major feature of the classification, it is now able to avoid the information loss and increase the features of the short text. Meanwhile the classifier could achieve soft output, in order to give a more accurate quantized value to term importance.

Keywords Classification, Dependency parsing, Term-weight, Query analysis, Term importance, Search engine, Information retrieval

2012年第30次中国互联网发展状况统计报告[1]指出,截止至2012年6月底,中国网民数量达到5.38亿,其中搜索引擎用户规模达到4.29亿,较2011年底增长2121万人,在网民中的渗透率攀升为79.7%,依旧是仅次于即时通讯的第二大网络应用。搜索引擎作为互联网的基础应用,是网民在互联网中获取信息的重要工具,其用户规模仍会随着网民总体规模的增长而进一步提升。然而,大部分搜索引擎用户只是普通的网络用户,在检索策略和检索技巧上都缺乏必要的知识,无法把自己想要寻找的查询信息用准确的关键词组合或者用规定的与或式等其他格式表达出来,用户提交给搜索引擎的查询往往是一些无规则的自然语言,带有很多的干扰信息。因此,对用户的Query中的term进行重要性识别,并对用户Query中的term重要性进行量化具有很重要的现实意义。

在搜索引擎和信息检索中,对用户的搜索请求进行分析是提高搜索引擎相关性的重要方向。在本文接下来的讨论中,将用户对搜索引擎的一次查询输入分词后的集合记为Query,将分词后得到的单词记为term。每一个term的重要程度用term-weight来标识。如何识别用户Query中term的重要程度是查询分析的重要组成部分,通过区分Query中term的重要程度,可以把用户的输入转化为让搜索引擎更容易理解的结构进行检索,并根据用户的输入来进行反馈,以帮助用户进一步明确自己的搜索目的和方向。因此,在信息检索中,给用户Query中的term赋予合适的权重,对其进行重要性识别并找出其中的关键短语对检索结果精度的提高有着极其重要的作用,它不仅能够提高检索结果的相关性,同时还可以很好地帮助查询分析和其它检索应用,比如查询中实体的识别[2]、查询的扩展[3,4]以及查询的翻译[5]等等。除此之外,关键短语提取还可以用来辅助查询推荐和目录搜索。因此,如何高效准确地识别term的重要程度有着重大的现实意义。

到稿日期:2013-01-15 返修日期:2013-04-02 本文受国家自然科学基金(70971059),辽宁省创新团队项目(2009T045)资助。
邱云飞(1976—),男,博士,教授,主要研究方向为数据挖掘理论及其应用;鲍 莉(1989—),女,硕士生,主要研究方向为数据挖掘;邵良杉(1961—),教授,博士生导师,主要研究方向为数据挖掘。

1 相关工作

在现有的搜索引擎和信息检索中，用户 Query 中 term-weight 通常是由一种上下文无关的方式得到的，这些方法都采用词袋思想，并没有考虑 Query 中 term 之间的相关性，如布尔模型、向量空间模型和概率模型等。如何给用户 Query 和文档中的 term 赋予适当的权重，并找出其中的关键短语，是提高信息检索准确性的关键所在。在工程实践中大部分系统用 IDF（Inverse Document Frequency）来表示 term 的 weight，通过统计频率而非有监督的机器学习方法来区分 term 的重要程度。

很早以前 Callan，Croft 和 Broglio[6] 就已经开始在信息检索的 Query 中进行关键词提取，随后 Allan[7] 等则确定了核心的 term 必须在相关文档中出现这一规则。Bendersky 和 Croft[8] 提出了从语言类查询中提取关键短语的有监督的方法，然而该方法限定了提取关键词短语的词性，只能从名词短语中提取关键短语。在文献[9]中，Kumaran 和 Allan 提出了对于长查询找出最优的子查询的方法，但是他们只是通过交互设置给出子查询建议而不是自动地确定最优的子查询。随后 Kumaran 和 Carvalho[10] 使用一个排序框架来自动地寻找最优的子查询，在排序过程中，他们综合了多种查询预测指标，比如互信息、查询清晰指数等等，但是他们的方法没有把注意力集中到 term 的 weight 上面。Lease[11] 等通过对一个用户 Query 的各种子查询在搜索引擎所产生的平均精度改变来得到最优的权重，当一个 term 缺失时的子查询获得最大的平均精度改变时，该 term 的权重最大。他们所采用的 Query 是比较长的 Query，而我们所用的用户 Query 比较短，一般只有 2～4 个 term，每个 term 的缺失都可能导致平均精度改变非常大。

本文提出了一种有监督的机器学习方法来学习用户 Query 中的 term-weight。该方法采用分类的方法，并引入了句法分析，把句法分析的结果作为分类的一项非常重要的特征来训练模型，考虑了用户 Query 中 term 之间的关系，既避免了由 Query 到单个 term 的信息丢失，又增加了短文本的特征，同时使分类器实现软输出，为 Query 中的 term 赋予合适的权重，能够给 term 的重要程度一个确定的量化值。

2 句法分析

句法分析有两大类型：短语句法分析和依存句法分析。它们分别对应目前应用比较广泛的两种语法体系：短语结构语法和依存语法。短语结构语法是终结符、非终结符、开始符号和重写生成式规则集的四元组，一个短语结构语法所接受的语言是由它的生成式规则所能导出的所有终结符串的集合；依存语法则用词之间的支配和被支配关系来描述语言结构，与短语结构语法相比没有非终结符，结构简单，更重要的是依存句法是直接按照词语之间的依存关系工作，是天然词汇化的，它不过多地强调句子中的固定词序，因此对自由语序的语言分析更有优势，其形式化程度较短语结构语法浅，对句法结构的表述更为灵活。依存句法的以上优势有助于提取用户 Query 的关键信息（关键词、关键短语等）及理解 Query 句法结构，因此在本文的方法中选择依存句法分析来分析句法结构。

2.1 依存句法相关算法

在基于统计依存句法分析中，主要有两种分析方法：一种是基于转移的分析方法，另一种是基于图的方法。两种方法在模型学习、分析方法以及搜索效率方面的比较如表 1 所列。

表 1 依存句法算法比较

	基于转移的算法	基于图的算法
学习过程	根据历史上转移的实例，学习一个状态转移的模型	学习一个为整个依存图打分的模型
分析过程	根据转移模型，构建一个优化的转移序列	根据模型，构建一个分数最高的依存图
搜索方式	贪心搜索，效率高	全局搜索，效率低

基于图的方法采用全局搜索策略，性能较高，但是效率却低；基于转移的方法采用贪心搜索策略，后续转移依赖于之前的转移序列，因此无法防止错误的繁衍，在性能上略低，但是基于转移的算法只需一遍扫描，具有线性时间复杂度，效率非常高，并且通过丰富的特征可以取得与基于图的方法相当的效果。考虑到性能与效率上的折衷，选择基于转移的句法分析方法。

在基于转移的句法分析方法中，通常有两种算法：一种是 Standard 算法，另一种是 Eager 算法，两者原理如表 2 和表 3 所列。表中输入 Query 为：$S=(w_0, w_1, \cdots, w_i, \cdots, w_j, \cdots, w_n)$，$\sigma$ 为单词的堆栈，β 为尚未处理的单词缓冲区序列，A (w_i, r, w_j) 为由 w_i 指向 w_j 的依存弧，依存关系为 r。在训练或解码中，当前三元组 (σ, β, A) 记为一个状态 state。

表 2 Arc-standard 算法

Transition	Action	Precondition
$(\text{Left-Arc})_r$	$(\sigma\|w_i, w_j\|\beta, A)\Rightarrow(\sigma, w_j\|\beta, A\cup\{(w_j, r, w_i)\})$	$i\neq 0$
$(\text{Right-Arc})_r$	$(\sigma\|w_i, w_j\|\beta, A)\Rightarrow(\sigma, w_i\|\beta, A\cup\{(w_i, r, w_j)\})$	
Shift	$(\sigma, w_i\|\beta, A)\Rightarrow(\sigma\|w_i, \beta, A)$	

表 3 Arc-eager 算法

Transition	Action	Precondition
$(\text{Left-Arc})_r$	$(\sigma\|w_i, w_j\|\beta, A)\Rightarrow(\sigma, w_j\|\beta, A\cup\{(w_j, r, w_i)\})$	$(w_k, r', w_i)\notin A$, $i\neq 0$
$(\text{Right-Arc})_r$	$(\sigma\|w_i, w_j\|\beta, A)\Rightarrow(\sigma\|w_i\|w_j, \beta, A\cup\{(w_i, r, w_j)\})$	
Reduce	$(\sigma\|w_i, \beta, A)\Rightarrow(\sigma, \beta, A)$	$(w_k, r', w_i)\in A$
Shift	$(\sigma, w_i\|\beta, A)\Rightarrow(\sigma\|w_i, \beta, A)$	

研究者在实验中已经指出，对于中文依存句法分析，Arc-standard 算法更加有效，我们设计采用 Arc-standard 算法。

2.2 依存句法训练及解码设计

依存句法系统主要完成两大功能：模型训练以及依存句法树解码。根据系统功能，将系统划分设计为两个大模块，即训练模块与解码模块。其中训练模块由特征提取模块、实例生成模块、分类器训练模块构成；解码模块由分类器分类模块、解码算法、依存树构造模块构成，也会用到特征提取模块。

分类器使用开源的机器学习工具包，包括张乐博士的最大熵工具包、台湾大学林智仁（Chih-Jen Lin）博士等开发设计的开源工具包 LibSVM 以及 LibLinear，考虑到速度与效果的折中，优先使用 LibLinear。

2.2.1 训练模块设计

训练模块主要由以下几个模块组成：

1. 特征提取模块：主要包括解析特征配置文件、特征抽取以及特征向量的生成等功能。此模块由模型训练、解码两大部分所共享。此模块需要实现的功能有：

1）特征选择：要对训练语料进行特征提取，首先要选择特征。我们选择的特征参照了 Nirve 的 MaltParser[12] 所用到的特

征，并以此为基础，增加了组合特征。具体如表4所列，表中STACK表示单词堆栈，BUFFER表示未处理的单词序列，LDEP与RDEP表示左、右孩子结点，DEPPREL表示依存类型。

2)特征计算：根据特征模板抽取出相应的特征，对特征计算特征值。

3)特征向量：将所有特征值进行组合，构成特征向量。

表4 特征示意图

简单特征	组合特征
STACK[0]_FORM	STACK[0]_POSTAG+BUFFER[0]_POSTAG
BUFFER[0]_ FORM	STACK[0]_CPOSTAG+BUFFER[1]_CPOSTAG
BUFFER[1]_ FORM	STACK[1]_CPOSTAG+BUFFER[0]_CPOSTAG
STACK[0]_CPOSTAG	STACK[0]_FORM+BUFFER[1]_CPOSTAG
BUFFER[0]_CPOSTAG	STACK[0]_FORM+BUFFER[0]_CPOSTAG
BUFFER[1]_CPOSTAG	BUFER[0]_CPOSTAG+BUFFER[1]_CPOSTAG+BUFFER[2]_CPOSTAG
LDEP(STACK[0])_DEPREL	……
RDEP(STACK[0])_DEPREL	
RNDEP(STACK[0])_DEPREL	
LDEP(BUFFER[0])_DEPREL	
RDEP(BUFFER[0])_DEPREL	
……	

2.实例生成模块：主要功能是从原始树库中生成训练实例。其中原始树库为CoNLL格式语料，训练实例的格式为：[类别标签(转移动作类型)特征向量]。此模块调用特征提取模块进行特征抽取，并根据一系列规则获取当前状态的转移类型，从而得到训练实例。一个实例由两部分组成，形式为[类别标签(transition action)特征向量(feature-vector)]，action为arc-standard的转移(Transition)动作，即left-arc，right-arc，shift。每个实例对应一个状态，状态为一个三元组(σ,β,A)的当前情况，其中σ为单词的堆栈，β为尚未处理的单词缓冲区序列，$A(w_i,r,w_j)$为依存弧，由w_i指向w_j，依存关系为r，在训练或解码中，当前三元组(σ,β,A)称为一个状态state。而动作action是根据堆栈、单词序列及所形成边和依存类型得出。

3.分类器训练模块：以训练实例为输入进行分类器模型训练。

2.2.2 解码模块设计

解码模块是对输入的文本进行依存分析，最后输出结果。本模块由以下几部分组成：

1.特征提取模块：对输入文本进行特征提取，与训练模块相同。

2.分类器预测模块：根据当前的状态特征进行动作(action)的预测，与训练过程类似，不同之处在于训练时根据规则确定动作，解码时根据分类器模型预测动作。

3.解码算法：转移算法的核心所在，负责对句子进行解码，并生成一棵无交叉的规范依存树。

4.依存树生成模块：根据解码生成依存树，最后输出。

3 基于SVM分类的term重要性识别方法

3.1 分类算法及其特征选择

对于分类系统而言，使用何种分类算法是分类系统的核心属性，对于本文而言，分类器训练的一个难点是用户Query非常短，平均长度一般是2～3个term，因此每个查询提交给分类器的特征非常少，即使分类器能组合和平衡特征库的优缺点，但对于任何查询的分类只能决定于少量的非空特征值。针对用户Query的这种特征，我们选择了对训练语料要求不多，在特征比较少时分类效果也不错的SVM(Support Vector Machine)分类算法，并选用林智仁博士的LibSVM进行训练和分类。分类方法可以归结为根据待分类数据的某些特征来进行匹配，选择最优的匹配结果，从而完成分类。因此核心的问题便转化为用哪些特征表示用户Query才能保证有效和快速的分类。最终我们选择的分类特征如下：

Query的长度：Query的长度为Query中term的个数，一般而言，短的Query中所有的term很可能都是重要的，而长的Query中往往包含一些不重要的虚词。

IDF：本文的方法没有采用经典的TF*IDF，是因为在检索系统中，Query中同样的term出现在不同的位置中会被当作两个不同的term来处理，例如"南京，南京"，IDF采用如下方式计算：

$$IDF(w_i)=\log\left(\frac{N}{\max(DF(w_i),\min_{k\in V}(DF(w_k)))}\right)$$

式中，N为用户Query的总数，V为用户Query日志中所有term的集合，IDF的统计语料来源于easou搜索引擎一年的用户Query，约50亿；

Term的词性：由词性标注得到，我们认为在用户Query中，有一些词性的词可能会更加重要，例如：名词往往比介词更加重要。

Term的offset：主要用来表征term在Query中的位置，基本单位是基本词的个数。

是否百科词条：该项特征主要是为了替代实体名称，因为常规的实体名识别在准确率上表现都不太好，同时对速度的影响较大，所以直接采用百科词典来代替实体名的识别，在准确率和速度上都能够得到保证。假如一个term是实体名称的一部分，那么如果认为这个term不重要而将其丢弃，必然会对搜索结果造成比较坏的影响，对百度百科进行抓取，共得到500万词条。

是否停用词：停用词指的是在搜索引擎的查询中对文档和Query的表征意义很低的词，例如"的"、"呢"等词汇。在本文的实现中，使用一个手动构建的停用词表来确定一个term是否为停用词，并且为了避免由于停用词出现问题而导致最终结果变差，在停用词的收录上比较严格，最终选定的停用词总量只有122个。

左右熵：左(右)熵表示该term与其左(右)边term之间的互信息(PMI)，PMI采用如下方式计算：

$$PMI(w_i,w_j)=\log\frac{p(w_i,w_j)}{p(w_i)p(w_j)}$$

式中，$p(w_i,w_j)$为w_i和w_j在用户查询日志中同时出现的概率，$p(w_i)(p(w_j))$为$w_i(w_j)$在用户查询日志中出现的概率，统计语料为easou搜索引擎一年的用户Query，并对其进行一定的处理，去除一些会对统计造成干扰的Query，包括URL、电话号码、用户账号(例如QQ号)、过长的Query(超过76个汉字)等，最终抽取处理后的1亿条未经过去重的Query来计算左右熵。

Term的依存关系：该项特征是句法分析模块的分析结

果，能够表征该 term 在整个 Query 语句中的依存关系，代表了其他 term 和整个 Query 对该 term 的依赖，在使用 term 的依存关系时进行了一些过滤，选择了一些比较重要的依存关系，如根节点依存关系、并列关系、主谓关系等等。

单个 term 为一个 Query 的概率：我们认为一个 term 成为一个 Query 的概率越大，说明这个 term 可能越重要，单个 term 为一个 Query 的概率采用如下方式计算：

$$STQR(w_i)=\frac{M}{N}$$

式中，M 为该 term 单独出现为一个用户 Query 的次数，N 为该 term 在用户 Query 中出现的总次数，统计语料为 easou 搜索引擎一年的用户 Query，约 50 亿。

3.2 结果反馈及修正

在前期的特征选取和分类模型训练中，由于人工标注的主观性和语料选取的片面性，很难保证所有的 term 重要性识别都能够做到完美，因此需要在发现错误报告后，及时地将正确的信息加入到训练语料中，然后定期地重新运行特征抽取模块和分类训练模块，建立流畅的错误反馈流程，使得 Query 中的 term 重要性识别不断地进步。

3.3 Term 重要性识别流程

Query 中 term 重要性识别流程如图 1 所示，其具体步骤如下。

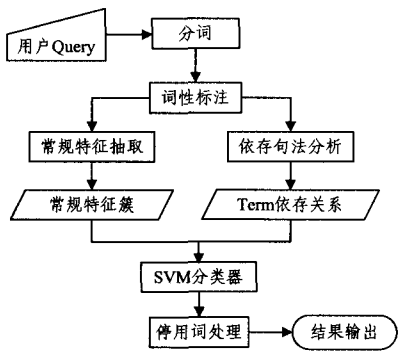


图 1 Query 中的 term 重要性识别流程

步骤 1 对长 Query 进行截断，纠错识别，包括以字为单位的 n-gram 方式进行的同音字纠错和拼音纠错，对于置信度较高的纠错，直接用纠错后得到的 Query 进行后面的分析，否则，将原 Query 和纠错后的 Query 同时进行 term 的重要性识别，并将得到的结果送入检索程序进行检索，将两种检索结果进行比较，最终把较好的结果在终端显示给用户；

步骤 2 对用户 Query 进行分词、词性标注，分词及词性标注均采用 easou 自有的分词库，其中分词的准确率为 99.1%，词性标注的准确率为 98.3%；

步骤 3 常规特征的抽取及其特征值的计算，包括 Query 的长度、IDF、term 的词性、term 的 offset、是否停用词、左右熵以及单个 term 为 Query 的概率等，特征的具体抽取方法在特征说明时已经描述；

步骤 4 使用本文所提到的句法分析方法对用户 Query 进行依存句法分析，得到每个 term 的依存关系，并对依存关系进行过滤，将最终得到的依存关系和步骤 3 的结果一起转化为 SVM 分类器所接受的输入格式作为 SVM 分类器的输入；

步骤 5 在步骤 3 和步骤 4 结果的基础上使用训练好的 SVM 分类器对 Query 中的每一个 term 进行重要性识别，其中 SVM 分类器利用人工标注好的数据以及步骤 3 和步骤 4 中的模块来进行训练；

步骤 6 对 SVM 分类器的分类结果进行停用词处理，对特殊的停用词的权重和重要性进行简单的调整；

步骤 7 输出最终结果。

4 实验设计及结果分析

4.1 数据标注

easou 搜索引擎 12 个月的用户查询日志是按照用户 Query 出现的频率由大到小来进行存储的，为了提高抽样的精度和确保样本的代表性，本方法选择了分层抽样。一共抽取了 2000 条长度大于 1 的用户 Query，其中 Query 长度为 2，3，4，5，6，以及 7 个以上的个数分别为 230，247，706，455，228 和 134。把 2000 个用户 Query 均分成 5 个子集，每个子集有 400 个用户 Query，每个子集中不同长度的 Query 是均等的，其中的 4 个子集作为训练集，1 个子集作为测试集，测试集中 Query 长度为 2，3，4，5，6，和 7＋的个数分别为 46，49，141，91，46 和 27。我们选择 15 个标注者对样本进行标注，确保每个用户 Query 都有 3 个标注者对其进行标注。

把用户 Query 中的 term 按以下 4 个级别进行标注：不重要的(Unimportant)、重要的(Important)、必需的(Required)、超级重要的(Super-important)，其对应的权重分子分别为 0、0.3、0.7 和 1。首先，将能够表明用户 Query 原始意义的 term 标记为 Important；其次，把停用词和一些丢弃后并不损失 Query 意义的 term 标记为 Unimportant；然后，将检索结果网页中相关性非常好的 term 标注为 Required；最后，将一旦丢弃就彻底改变用户 Query 意义，也就是绝对不能丢弃的 term 标记为 Super-important。如果 3 个标注者对同一个 Query 标注的类别不一样，则该 term 的权重分子取其均值。

4.2 评价标准

我们使用下面的指标来直接衡量机器学习方法的好坏：

1)肯德尔相关系数(KTCC)：衡量机器学习方法所得出的 term 重要程度的序列与人工标注所得的序列的一致程度，值为 1 时表示完全一致，值为－1 时表示完全相反，而值为 0 时表示两个序列没有关系，是相互独立的；

2)皮尔森相关系数(PCC)：衡量机器学习方法所得出的 term-weight 与人工标注所得出的 term-weight 的相关程度，值为 1 时最好，表示与人工标注完全一样；

3)最重要词的准确率(TOP-1)：衡量在用户 Query 中最重要的词的准确率。

给用户 Query 中的 term 赋予合适的权重，判断其重要程度，并进而找出其中的关键 term，最终的目的是提高检索结果的相关性。Query 中 term 重要性和权重分子是 query 分析的核心模块，我们把它嵌入到 Query 分析模块中进行实验，在 easou 搜索引擎中其他模块均保持不变的情况下，分别对我们最终提出的引入句法分析后的分类方法(方法 1)、引入句法分析前的分类方法(方法 2)以及经典的基于 IDF 方法(方法 3)的检索结果进行评测。采用目前比较流行的相关性打分方法[13]，只考虑检索结果网页与 Query 的相关性，不考虑网页质量、是否权威等因素，对检索结果的前 3 条进行评测。评测以方法 1 为基准，计算方法 1 与方法 2 以及方法 1 与方法 3

的结果胜出率、胜出和均分胜出率，其中胜出率$=\frac{a+b}{a+2b+c}$，胜出$=\frac{a-c}{a+2b+c}$，均分胜出率$=\frac{m-n}{m}$。其中a为方法1比方法2(方法3)好的个数，b为方法1与方法2(方法3)一样好的个数，c为方法1没有方法2(方法3)好的个数，m为方法1的综合平均得分，n为方法2(方法3)的综合平均得分。

4.3 结果分析

表5显示了使用本文所提出的方法对用户Query“北京到仰光的时差”和“去长江游泳”进行学习的结果，对于这两个用户Query，本文所提出的基于分类的机器学习方法对其中term的重要性的识别与人工标注的结果均是完全一致的。

表5 学习term重要性的例子

查询词	人工标注		方法1	
	排名	分值	排名	分值
北京	3	0.8000	3	0.6785
到	4	0.1000	4	0.0513
仰光	2	0.9000	2	0.7129
的	5	0.0000	5	0.0247
时差	1	1.0000	1	0.7468
去	3	0.1000	3	0.0852
长江	2	0.7000	2	0.6266
游泳	1	0.9000	1	0.7357

为了得到单个特征的重要程度，每次只用一个特征来训练模型。将最重要的6个特征在表6中列出，可以看到IDF单独作为特征训练时，其KTCC系数最高，因此我们可以把IDF作为一个合理的内在评价的基准；其他比较重要的特征有term的依存关系、单个term为Query的概率、是否停用词、term的词性以及左右熵等。请注意，有条件的特征重要性可能导致特征的重要程度排序完全不同，特别是当这些特征是相关的时候，所以我们训练模型时采用所有我们所选择的特征而不仅仅是那些排名靠前重要的特征。表7表明我们所选用的方法与经典的IDF方法相比，其肯德尔相关系数、皮尔森相关系数以及最重要词的准确率都要高很多，说明我们所采用的方法在term的重要性识别上要比IDF方法好；在引入句法分析的结果后比引入句法分析前的方法要好，说明句法分析是一个很重要的特征，考虑了用户Query本身所带的信息。

表6 在测试集中单个特征的表现

排名	特征	KTCC
1	IDF	0.3178
2	term的依存关系	0.3112
3	单个term为query	0.2943
4	停用词	0.2371
5	词性	0.2126
6	左右熵	0.1780

表7 算法比较

度量标准	方法1	方法2	方法3
KTCC	0.4207	0.3714	0.3356
PCC	0.6678	0.6189	0.5177
TOP-1	0.6721	0.6213	0.6142

从表8和表9中可以看出方法1与方法2和方法3相比均有一定程度的提升，其中方法1较之方法3的提升更大，方法2较之方法3也有一定程度的提升。提升的主要原因均是在对Query中的term进行重要性识别中引入了更多正向特征。当Query中的term在7个或7个以上时，方法1较之方法2对效果的提升有一定的降低，原因在于当Query的复杂度达到一定的程度时，整个搜索引擎体系相关性品质的不足开始显现，相关性得分的降低不再单一地由term重要性不合理引发，有可能是天然的数据不足、查询类别等因素引起。在7个以下term的Query中，2－4个term的Query在引入句法分析这一特征后没有5－6个term的Query提升明显，其原因在于：2－4个term的Query有一部分是由单纯的关键词组合成的，在句法分析时会出现误差，方法1会退化成方法2，所以对于效果的提升没有起到更好的作用。

表8 方法1和方法2检索结果对比

Query长度	总计	方法1>方法2	方法1=方法2	方法1<方法2	胜出率	胜出	方法1综分均分	方法2综分均分	均分胜出率
2—4	236	89	96	51	55.72%	11.44%	3.32	3.23	2.71%
5—6	137	66	39	32	59.66%	19.32%	3.15	3.03	3.81%
7+	27	12	7	8	55.88%	11.76%	2.98	2.92	2.01%
总计	400	167	142	91	57.01%	14.02%	3.24	3.14	3.09%

表9 方法1和方法3检索结果对比

Query长度	总计	方法1>方法3	方法1=方法3	方法1<方法3	胜出率	胜出	方法1综分均分	方法3综分均分	均分胜出率
2—4	236	97	85	54	56.70%	13.39%	3.32	3.17	4.51%
5—6	137	72	32	33	61.53%	23.06%	3.15	3.00	4.76%
7+	27	11	9	7	55.55%	11.11%	2.98	2.92	2.01%
总计	400	180	126	94	58.17%	16.34%	3.24	3.09	4.63%

结束语 本文采用基于分类的有监督的机器学习方法对用户Query中term-weight进行学习，并引入了句法分析，考虑了用户Query中term之间的关系，既避免了由Query到单个term的信息丢失，又增加了短文本的特征。给用户Query中的term赋予合适的权重，可以判断term的重要程度，并找出其中的关键短语，进而提高信息检索的准确率和召回率，同时还改善了搜索引擎的用户体验。在以后的工作中，我们会针对如何根据term的权重和该term在Query起到的作用来进行信息检索中Query变换的研究。

参考文献

[1] 第30次中国互联网发展状况统计报告[R]. 中国互联网络信息中心(CNNIC)，2012

[2] Guo Jia-feng, Xu Gu, Chen Xue-qi, et al. Named entity recognition in query[C]// Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval. Boston, MA, USA: ACM, 2009: 267-274

[3] Fonseca B M, Golgher P, Possas B, et al. Concept-based interac-

tive query expansion[C]//Proceedings of the 14th ACM international conference on information and knowledge management. New York, NY, USA: ACM, 2005: 696-703

[4] Cao G, Nie J Y, Gao J, et al. Selecting good expansion terms for pseudo-relevance feedback[C]//Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA: ACM, 2008: 243-250

[5] Gao J, Nie J Y, Xun E, et al. Improving query translation for cross-language information retrieval using statistical models[C]// Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA: ACM, 2001: 96-104

[6] Callan J P, Croft W B, Broglio J. Trec and tipster experiments with inquery [C]//Information Processing and Management: an International Journal-Special issue: the second text retrieval conference(TREC-2). 1995: 327-343

[7] Allan J, Callan J, Croft W B, et al. Inquery at trec-5 [C]// TREC. 1997: 119-132

[8] Bendersky M, Croft W B. Discovering key concepts in verbose queries[C]//Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA: ACM, 2008: 491-498

[9] Kumaran G, Allan J. Effective and efficient user interaction for long queries[C]//Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA: ACM, 2008: 11-18

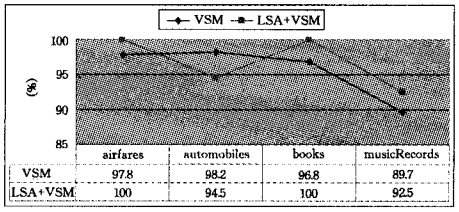
[10] Kumaran G, Carvalho V R. Reducing long queries using query quality predictors[C]//Proceedings of the 32nd annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA: ACM, 2009: 564-571

[11] Lease M, Allan J, Croft W B. Regression Rank: Learning to Meet the Opportunity of Descriptive Queries[C]//Proceedings of the 31st European Conference on IR Research on Advances in Information Retrieval. Toulouse, France, 2009: 99-101

[12] Nivre J, Hall J, Nilsson J. MaltParser: A data-driven parser-generator for dependency parsing [C]//Proc. of LREC. 2006

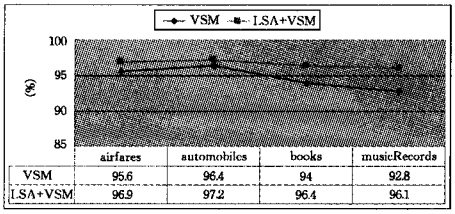
[13] 李珏伶. 搜索引擎网页相关性评估方法设计及其在 rank 模型上的应用[D]. 北京: 北京交通大学, 2011

(上接第 230 页)



	airfares	automobiles	books	musicRecords
VSM	97.8	98.2	96.8	89.7
LSA+VSM	100	94.5	100	92.5

图 4 查全率 R 对比



	airfares	automobiles	books	musicRecords
VSM	95.6	96.4	94	92.8
LSA+VSM	96.9	97.2	96.4	96.1

图 5 F1 值对比

对比实验结果可以得出：(1)使用 LSA 后，查准率较单纯使用 VSM 有所提高。(2)对于存在领域属性重叠的查询接口，采用 LSA 后，反而使部分并不属于同一领域的查询接口之间的相似度增大，本实验中，由于 Automobiles 领域和其他的 3 个领域存在属性重叠现象，由图 4 可以看出，使用 LSA 后查全率反而降低。(3)综合考虑查准率 P 和查全率 R，从 F_1 值可以看到使用了 LSA 后，整体聚类质量有了明显提升。

结束语 本文提出一种基于潜在语义分析的 Deep Web 查询接口聚类方法，新方法通过引入 LSA 理论，发现了 Deep Web 查询接口之间的隐含的语义关系，提升了领域内查询接口之间的相似度。通过在 UIUC 的 Web 集成资源库提供的查询接口集上进行实验，验证了本文提出方法的有效性，相较于传统的基于向量空间模型的聚类方法，新方法明显地提升了聚类质量。

本文提出的新方法仍有值得讨论的问题，如在数据预处理阶段，本文是通过人为地定义特征词表来进行语义拓展，因此在下一步研究工作中，我们将考虑如何进行本体学习，将领域本体引入进来，尝试在本体的指导下进行查询接口的语义拓展，进一步提高 Deep Web 查询接口聚类的效果。

参 考 文 献

[1] 刘伟，孟小峰，孟卫一. Deep Web 数据集成研究综述[J]. 计算机学报，2007，30(9)：1475-1489

[2] Olney, Andrew M. Generalizing Latent Semantic Analisis[C]// 2009 IEEE International Conference on Semantic Computing. 2009: 40-46

[3] Liu Yun-feng, Qi Huan. Latent Semantic Analysis of Chinese Information[J]. Journal of South China University of Technology (Natural Science), 2004(32): 107-111

[4] Li Ya-xiong, Zhang Jian-qiang, Dan Hu. Text Clustering Based on Domain Ontology and Latent Semantic Analysis[C]//2010 International Conference on Asian Language Processing. 2010: 219-222

[5] Thomas H. Unsupervised Learning by Probabilistic Latent Semantic Analysis[J]. Machine Learning, 2001, 42(2): 177-196

[6] 黄承慧，印鉴，侯昉. 一种结合词项目 TF-IDF 方法的文本相似度量方法[J]. 计算机学报，2011，34(5)：857-864

[7] Mao Qin-jiao, Feng Bao-qin, Pan Shan-lang. Latent Semantic Analysis for Query Ierfaces of Deep Web Site[J]. Journal of SouthEast University (English Edition), 2008, 24(3): 312-314

[8] 盖杰，王怡，武港山. 基于潜在语义分析的信息检索[J]. 计算机工程，2004，30(2)：58-60

[9] Wu Chen, Vidyasagar P, Chang E. Latent Semantic analysis-The Dynamics of Semantics Web Services Discovery[J]. Lecture Notes in Computer Science, 2008, 4891: 346-373