基于依赖关系的定义类问题回答系统

曹均阔 [1, 2] 黄萱菁 [1]

摘 要 定义句模板和关联词向量作为候选定义句选择的常用特征, 前者确定定义的表达形式, 后者确定定义的叙述内容. 目前, 大多数定义类问题回答系统中都是根据问题 Target 及其周边词之间的相对位置关系来提取定义句模板和关联词. 在这种基于位置关系的基础上提出了基于依赖关系的定义句模板和关联词抽取方法, 然后将这些特征应用到改进的在线算法 MIRA (Margin infused relaxed algorithm), 从而实现对候选定义句子的排序. 这种改进的 MIRA 算法能够根据学习进程自动调整约束条件, 从而提高算法的收敛速度与性能.

关键词 定义类问题回答, MIRA, 依赖关系, 在线算法
中图分类号 TP391

A Definitional Question Answering System Based on Dependency Relation

CAO Jun-Kuo[1, 2] HUANG Xuan-Jing[1]

Abstract Definitional pattern and centroid word vector are the popular features for definitional question answering. The former determines how to describe the information, and the latter determines what topics should be concerned. Most current systems use sequence pattern and sequence centroid words extracted only by relative position to question target to identify definition sentences. In contrast to sequence knowledge, we propose dependency-based knowledge, including dependency pattern and dependency centroid word extracted by dependency relation to question target. Specifically, we use the improved ultraconservative online algorithm and margin infused relaxed algorithm (MIRA) for the task of candidate sentences ranking. Experiments have demonstrated that the improved MIRA can greatly increase the system performance in terms of progressive reduction of optimization constraints as training moves forward.

Key words Definition question answering, margin infused relaxed algorithm (MIRA), dependency relation, online algorithm

定义类问题回答作为国际文本信息检索会议 (Text retrieval conference, TREC) 的一项重要评测任务, 越来越受到人们的广泛关注[1−2]. 这类问题回答类似于: 请告诉我 “X” 的相关信息, 其中 X 是问题目标词 (Target). 例如: 什么是 “库尔斯克号” 事件? 返回的答案应该包括: 失事时间、地点、大致原因、营救措施以及相关的重要信息. 目前, 在 TREC 会议的这类回答问题中, 一般分为 3 步: 1) 将问题的 Target 作为 Query 从语料库中检索相关文档, 然后进行切句处理, 生成候选答案句子; 2) 将抽取的候选定义句子与 Target 进行相关度计算并进行排序; 3) 去除冗余后, 选择排名靠前的一些句子作为问题答案. 就广义的定义类问题回答而言, 应进一步考虑对返回答案句子进行组织与重述, 这个工作 TREC 会议并未作要求. 本文的实验主要针对第 2 步工作, 即候选句子排序. 排序任务通常分两步完成: 首先充分抽取有效特征, 然后将这些特征应用到排序算法或分类器. 在定义类问题回答系统中, 常用的有效特征有: 定义句模板和关联词向量[3−4]. 模板抽取已经被广泛应用到信息抽取任务. Sudo 等运用 TF × IDF 提取相关的句子, 并从这些句子中抽取定义句模板. 另外一些关于模板的文献则致力于模板匹配算法研究[4−5]. 这些算法利用人工规则或机器学习方法实现句子排序. 对于关联词在定义类问题回答中的使用, 通常是根据 Target 构建一个关联词向量, 然后计算候选句子与该向量之间的相似性, 用以选择问题的答案. 本文的研究工作主要是基于依赖关系的知识抽取, 包括定义句模板和关联词向量, 然后将这些特征应用到在线算法 MIRA (Margin infused relaxed algorithm), 从而实现对候选句子的排序. 一般来说, 在线算法在每一个学习周期致力于两个目标: 1) 调整当前分类器, 使之能正确地区分当前训练样例; 2) 尽量对当前分类器进行最小调节, 而不是大幅度改变, 以维护分类器的连续性和稳定性[6−7]. 文献 [7−8] 在介绍感知器的基础上, 系统地阐述了在线算法的学习流程, 并且首次提出了保守与超保守在线算法的概念. 国内对在线算法系

收稿日期 2008-07-17 收修改稿日期 2009-02-20
Received July 17, 2008; in revised form February 20, 2009
国家自然科学基金 (60435020, 60673038) 资助
Supported by National Natural Science Foundation of China (60435020, 60673038)
1. 复旦大学计算机学院 上海 200433 2. 南昌航空大学计算机学院 南昌 330036
1. School of Computer Science, Fudan University, Shanghai 200433 2. School of Computer Science, Nanchang Hangkong University, Nanchang 330036
DOI: 10.3724/SP.J.1004.2009.01429

统研究的文献尚不多见, 大多都致力于现有分类器 (如 SVM, Boosting) 引入在线学习规则, 从而克服传统分类器在海量训练数据及适时性上的不足[9−11]. MIRA 是一种超保守在线算法, 在分类、排序、预测等应用领域取得不错成绩[12−13]. 本文在传统 MIRA 算法基础上进行了改进, 提出 K-best MIRA 与动态 K-best MIRA 算法. 这种新型的 MIRA 算法能够根据学习进程自动调整优化约束条件, 从而提高算法的收敛速度与性能.

1 基于依赖关系的知识抽取

设 x 是任意一个由词组成的非空串, X 是由 x 所有词构成的集合. 定义在 X 上的任意二元关系 $\rightarrow$, 如果有向图 $\langle X, \rightarrow\rangle$ 是一棵树, 则二元关系 $\rightarrow$ 被称为依赖关系, 树 $\langle X, \rightarrow\rangle$ 被称为 x 的依赖树. 语言学界已经证明, 如果对句中任意具有句法关系的词都设定修饰者和被修饰者, 则建立在句子单词集合上的这种句法关系是依赖关系, 可以用依赖树来表示. 如句子 "HORUS, who was half-man, was worshiped by ancient Egyptians" 的依赖树如图 1 所示.

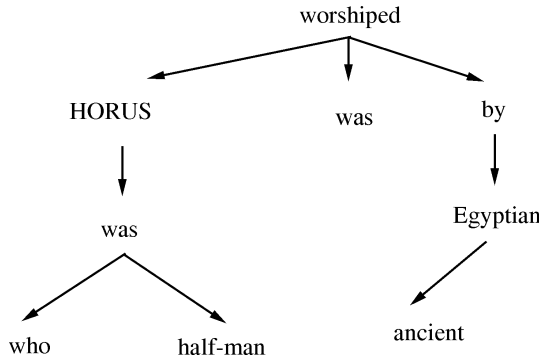

图 1 一棵简单的句法依赖树

Fig. 1 A simple dependency syntax tree

对于一个实体、概念或事件而言 (对应问题的 Target), 其定义通常是一组相关重要信息的描述型句子. 这些定义类型的句子从内容上往往围绕 Target 某个支点展开, 而与这个支点相关的词, 我们称之为 Target 的关联词. 由这些关联词组成的向量, 则称之为关联词向量. 从定义句的表现形式上来说, 往往蕴含某些特定的表达方式, 我们称这些表达方式为定义句模板. 本文将详细讨论如何利用依赖关系进行定义句模板和关联词向量的抽取.

1.1 定义句模板

1.1.1 模板抽取

我们以 TREC 官方公布的定义类问题答案为训练语料, 从中抽取蕴含的定义句模板. 仍然以定义句 "HORUS, who was half-man, was worshiped by ancient Egyptians" 为例, 显然该例句对应的 Target 是 HORUS. 文献 [4] 指出, 定义句模板通常由 Target 及其附近的词或标点符号决定. 基于此, 该文献提出一种基于顺序的模板, 其形式为 $\{token_{-2}, token_{-1}, Target, token_1, token_2\}$, 其中 $token$ 表示作者利用人工规则替换的一些 Tag 标记, $token_{-1}$ 和 $token_1$ 分别表示 Target 前后相邻的标记, 依次类推. 因此对应该例句抽取的顺序模板为: {null, null, $Target$, , , who}. 同样, 如果假设定义句模板通常由 Target 及其相邻的依赖关系决定, 则我们可以采取如下 3 个步骤抽取基于依赖的定义句模板:

1) 句法分析: 首先利用一个句法分析器, 对定义句进行句法分析, 从而得到一棵类似于图 1 的依赖树. 本文选用的句法分析器是 Minipar[14].

2) 关系系列抽取: 从 Target 出发抽取关系系列, 这个序列上至根节点, 下至叶子节点. 由于 Target 在句法树可能有多个儿子节点, 所以抽取的关系序列往往是一个集合. 从图 1 不难得到的关系系列有两个, 分别是: {who → was → HORUS → worshiped}以及 {half-man → was → HORUS → worshiped}.

3) 提炼模板: 类似于基于顺序的定义句模板, 我们只取 Target 的上下两级依赖关系, 其表现形式如 {grandpa, father, $Target$, child, grandchild}, 这个关系序列称之为初始模板. 同理, 考虑到初始模板的泛化能力不足, 我们亦作对应的标记替换. 利用表 1 规则替换后, 例句中的两个初始模板均对应相同的

表 1 词标记的启发性替换规则

Table 1 Manual heuristics rules used for token substitution

被替换词	替换标记	示例
问题 Target	Target 对应的 pos tagger, 并外加括号	HORUS→(N)
is, am, are, was, were, be	VBE	was→VBE
a, an, the	DT	the→DT
数字	CD	2008→CD
其他所有词	对应的 pos tagger	by→Prep

定义句模板: {N → VBE → (N) → V → null}.

1.1.2 模板匹配

模板匹配的方式有两种: 一种是硬匹配 (Hard matching), 另一种是软匹配 (Soft matching). 所谓硬匹配, 就是逐一比较模板之间的槽, 如果所有槽对应完全一致, 则认为模板匹配, 否则不匹配. 一般而言, 硬匹配的条件过于严格. 所以本文使用基于语言模型的软匹配方法. 语言模型作为自然语言处理的一个常用工具, 已经被广泛地应用到词性标准、句法分析、问题回答等任务中[15−16]. 限于训练数据的容量, 本文采纳一元语言模型 (Unigram) 和二元语言模型 (Bigram) 线形插值的方法. 假定定义句模板 $p = \{t_1, \cdots, t_l\}$, 计算其概率 $P(p)$ 使用如下公式:

$$P(p) = P(t_1, \cdots, t_l) = P(t_1|S_1) \times \prod_{i=2}^{l}(\lambda P(t_i|t_{i-1}) + (1-\lambda)P(t_i|S_i)) \tag{1}$$

其中, $P(t_i|S_i)$ 表示槽 S_i 出现标记 t_i 的条件概率, λ 是一元语言模型与二元语言模型的联合权重, l 表示模板的长度 (不包含 null). 另外, 考虑到某些标记的出现频率可能为零, 我们使用加 1 平滑处理.

1.1.3 参数 λ 估计

对于式 (1) 中联合权重 λ, 我们采用期望最大法 (Expectation maximization, EM) 进行估计 [5, 16]. 假定从训练预料中抽取的所有定义句模板为 n 个, 则期望最大估计可表示为

$$\lambda = \arg\max_{\lambda} \sum_{j=1}^{n} P(t_1^j, \cdots, t_{l_j}^j|\lambda) = \arg\max_{\lambda} \sum_{j=1}^{n} \frac{1}{l_j - 1} \sum_{j=2}^{l_j} \log(\lambda P(t_i^j|t_{i-1}^j) + (1-\lambda)P(t_i^j|S_i^j)) \tag{2}$$

其中, l_j 表示模板的长度. 由于 $P(t_1|S_1)$ 是一个常量, 并不影响对 λ 的估计, 所以并未列入式 (2) 计算. λ 的期望最大估计算法描述如下.

步骤 1. 初始化 λ, $\lambda \in (0, 1)$;

步骤 2. 利用下式更新 λ

$$\lambda = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{l_j - 1} \sum_{i=2}^{l_j} \frac{\lambda P(t_i^j|t_{i-1}^j)}{\lambda P(t_i^j|t_{i-1}^j) + (1-\lambda)P(t_i^j|S_i^j)} \tag{3}$$

步骤 3. 重复步骤 2, 直到 λ 收敛.

1.2 关联词向量

关联词向量在一些文献中亦称之为词袋 (Bag-of-words), 已经被广泛地应用到自动文摘系统[17] 和定义类问题回答系统中[3, 5]. 最早抽取关联词的方法大多采用统计与中心词同时出现次数 (Cooccurrence). 同现次数越多, 则与中心词关系越密切. 后来研究者发现, 这种统计同现次数的方法, 并没有考虑到中心词的上下文. 因此, 人们进一步提出基于上下文的统计同现词方法 (Local context analysis, LCA). 其中, 文献 [5] 是一种常见的基于 TFIDF 统计的 LCA 方法, 其关联度 $R(T, w)$ 计算方法如下

$$R(T, w) = -\log \frac{SF(T, w)}{SF(T)SF(w)} \times IDF(w) \tag{4}$$

其中, T 是中心词, w 为与 T 在同一句子中的同现词. $SF(T, w)$ 是同时包含词 T 和 w 的句子数, $SF(w)$ 是包含词 w 的句子数, 而 $IDF(w)$ 则表示出现词 w 的文档数倒数. 这种基于上下文的同现关系统计方法的确能较好地反映词与词之间的关联度, 但是这种方法并没有考虑到同现句子本身的重要性, 因此文献 [3] 将词与词关联度的计算引入了句子的重要性概念.

$$R(T, w) = \sum_{j} E(w, s_j) \times Score(s_j, T) \tag{5}$$

其中

$$E(w, s_j) = \begin{cases} 1, & \text{若 } w \in s_j \\ 0, & \text{若 } s \notin s_j \end{cases} \tag{6}$$

其中, $Score(s_j, T)$ 表示根据中心词计算的句子权重. 本文不同于文献 [3] 的地方在于: 将基于依赖关系的定义句模板应用到句子权重计算当中. 对于问题 Target, 我们首先检索返回文档集, 然后切句得到句子集合 $S = \{s_1, s_2, \cdots, s_m\}$. 进一步设定文档的初始权重为搜索引擎提供的分数. 这里我们默认搜索引擎已经对文档进行了较好排序, 相关度高的文档中出现的句子则与 Target 更相关. 给定任意句子 s, 我们根据问题 Target 从中抽取初始模板 $R = \{r_1, r_2, \cdots, r_q\}$, 且每个初始模板 r_i 对应定义句模板 p_i. 另外, 考虑到 w 可能在多个句子中出现, 所以词 w 与 Target 的关联度计算公式如下

$$R(T, w) = \sum_{j} Score_D(s_j|w \in s_j) \times Score_P(w) \tag{7}$$

其中, T 为 Target, $Score_D(s_j|w \in s_j)$ 表示包含词 w 的句子 s_j 在文档的权重. 实质上是 s_j 与 Target 相关度的简单度量, 计算模型有很多, 本文直接引用了搜索引擎返回的打分, 而词所对应模板分值定义如下:

$$Score_P(w) = \begin{cases} P(p_i), & \text{若 } w \in r_i \\ 0, & \text{若 } w \notin r_i \end{cases} \quad (8)$$

最后, 我们取关联度较高的关联词组成关联词向量.

2 MIRA 算法

Crammer 和 Singer 于 2003 年最早提出 MIRA 算法[7], 随后 MIRA 算法成功地应用于句子压缩[18]、句法分析[14, 19−20]、语义分析等自然语言处理问题[21]. 文献 [14, 19−20] 对 MIRA 算法进行了扩展, 用以实现对结构化数据分类, 改进后的 MIRA 算法如下:

算法 1 (改进后的 MIRA 算法).

假设训练数据: $(x_n, y_n)_{n=1}^{N}$

1 $w^{(0)} = 0$; $v = 0$; $i = 0$
2 **for** $m : 1, \cdots, M$
3 **for** $n : 1, \cdots, N$
4 $\min |w^{(i+1)} - w^{(i)}|$
5 s.t. $s(x_n, y_n) - s(x_n, y') > L(y_n, y')$
6 $\forall y' \in O(x_n)$
7 $v = v + w^{(i+1)}$
8 $i = i + 1$
9 **end for**
10 **end for**
11 $w = \dfrac{v}{M \times N}$

从上述算法中不难看出, M 表示迭代次数, N 表示训练样本的个数, 算法返回的结果是一个平均后的权重向量 $\boldsymbol{w}$, 而 $\boldsymbol{v}$ 则是用来累积存放每个训练周期后权重更新的辅助向量. Collins 在文献 [4] 中指出, 这种平均后的权重有利于减少由于过拟合造成的不良影响. 改进后的 MIRA 算法, 实质上是一个求解带约束条件的二次规划问题. 对于此类问题, 我们可以借助 Hildredth 算法或 Goldfarb 算法进行求解. 文献 [14, 19] 中的结构数据均是语法分析树, 所以 $s(x, y)$ 表示根据句子 x 得到句法树 y 的概率, 损失函数 $L(y, y')$ 反映的是 y 与 y' 所包含语法关系的差异程度. 同理, $O(x)$ 则是由 x 可能产生的所有句法树. 在改进后的 MIRA 算法中, 约束条件 (算法 1 的第 5, 6 行) 只纳入了训练样例 x 的所有输出 $O(x)$ 和目标 y, 而并未考虑当前学习周期下 $O(x)$ 和 y 的输出顺序. 一般情况下, 如果 $O(x)$ 是一个小数量级的集合, 这个约束条件是完全可行的. 反之, 如果 $O(x)$ 是一个大数量集的集合, 我们必须考虑以下两个问题: 1) 算法中的二次规划问题可能找不到合适解; 2) 即使找到解, 也往往容易产生过拟合现象. 因此, 在 MIRA 算法的基础上, 我们进一步提出 K-best MIRA 和动态 K-best MIRA 算法, 以克服上述问题.

定义 1. 对于当前训练样例 (x, y), 如果 MIRA 算法最多只取前 k 个的输出结果 $O_k(x)$ 作为约束条件, 而不是所有输出 $O(x)$, 则称该 MIRA 为 K-best MIRA (简称 K-MIRA).

定义 2. 对于当前训练样例 (x, y), 如果 MIRA 算法在所有输出 $O(x)$ 中, 从目标值 y 开始, 依次向前取 k 个输出结果作为 Margin 的约束条件, 则称该 MIRA 为动态 K-best MIRA (简称 DK-MIRA).

定义 1 与定义 2 的本质区别在于, 前者只考虑 $O(x)$ 输出顺序, 而后者同时考虑当前训练状态下 y 的输出位置. 本文将 K-best MIRA 算法用于定义类问题回答中的候选句子排序任务, 实验数据表明: k 取一个较小的常量即能获取较好的训练结果. 随着 k 的增大, 算法的性能会由于过拟合问题而逐渐下降. 显然, 对于参数 k 的设置将直接影响算法的泛化能力. 改进后的 MIRA 算法能够根据学习进程自动调整约束条件, 从而提高算法的收敛速度与性能. 从算法的时间复杂度来看, t 时刻 MIRA 的复杂度为: $\mathrm{O}\left(|O(x_t)| \times |Y_t|\right)$, 而 K-MIRA 与 DK-MIRA 的复杂度为: $\mathrm{O}(k \times |Y_t|)$, 这里 Y_t 表示 x_t 的目标. 显然, 当 x_t 的可能输出是一个大数量的集合时, K-MIRA 与 DK-MIRA 的速度优势将变得非常明显.

3 实验与数据

本文共设计了四组实验来评测我们的系统. 第一组实验是定义句模板的评测, 用来评测基于顺序的模板和基于依赖关系的模板, 以及评测模板硬匹配和软匹配方法; 第二组实验是关于关联词向量的实验, 用以比较常见关联词抽取方法的性能; 在第三组实验中, 我们利用相同的数据训练 MIRA, K-MIRA 以及 DK-MIRA, 从而验证 K-MIRA 以及 DK-MIRA 的有效性; 最后一组实验, 我们将 DK-MIRA 与当前 TREC 参评的几个优秀系统进行比较. 在前三组实验中, 其训练数据来自 TREC 2005, 测试数据来自 TREC 2006. 最后一组实验的训练数据来自 TREC 2004, 测试数据来自 TREC 2005. 另外, MIRA 系列算法中损失函数的参数阈值同文献 [13], 均设置为 1, 迭代步长为 50. 在详细介绍实验之前, 我们先简单地介绍一下定义类的问题回答评测方法[1−2]. 给定每一道问题, TREC 会议专家根据分析所有参评单位的递交答案后, 给出该问题的 Vital Nuggets 和 Okay Nuggets. 其中 Vital Nuggets 表示至关重要的信息, 要求系统必须返回; 而 Okay Nuggets 表示一般相关信息. TREC 官方根据系统提交答案信息, 计算最终的 $F(\beta = 3)$ 成绩为

$$F(\beta=3)=\frac{10\times P\times R}{9\times P+R} \tag{9}$$

其中, R 为 Vital Nuggets 的召回率, P 为精确率, 详细的评测方法参见文献 [1−2]. 根据 TREC 的评测标准, 文献 [22] 设计了一套比较权威的自动评测工具 “Pourpre v1.0c”. 为了评测的客观性, 本文所有实验评测均基于这套评测工具, 并且同文献 [22] 设置一样, 系统返回排名靠前的 12 个句子作为问题答案.

3.1　定义句模板评测

在这组实验中, 我们分别使用顺序模板和依赖关系模板进行定义类问题回答. 针对这两种类型的模板, 分别采用软匹配和硬匹配的方法对候选句子进行排序. 假设 P_{all} 为从训练数据中抽取的所有模板集合, 根据问题 Target, 候选句子 s 可抽取的模板为 $\{p_1,p_2,\cdots,p_q\}$, 则对应该句子硬匹配分值为

$$Score_{HP}(s)=\sum_i P(p_i|p_i\in P_{\text{all}}) \tag{10}$$

而对应该句子软匹配分值为

$$Score_{HP}(s)=\sum_i P(p_i) \tag{11}$$

从实验结果 (见表 2 所示) 不难看出, 无论是依赖关系模板还是顺序模板, 软匹配方法均好于硬匹配方法. 其中在顺序模板中, 软匹配比硬匹配提高了 0.002, 而在依赖关系模板中, 软匹配比硬匹配提高了 0.004. 同时, 无论是软匹配方法还是硬匹配方法, 依赖关系模板均好于顺序模板. 其中利用硬匹配方法时, 依赖关系模板比顺序模板提高了 0.005, 而利用软匹配方法, 依赖关系模板比顺序模板提高了 0.007.

表 2　定义句模板及匹配算法的比较

Table 2　The comparison of pattern and matching algorithms

定义句模板 + 匹配算法	F 成绩
顺序模板 + 硬匹配	0.301
顺序模板 + 软匹配	0.303
依赖关系模板 + 硬匹配	0.306
依赖关系模板 + 软匹配	0.310

3.2　关联词评测

关联词抽取的方法有很多, 本文分别对以下三种关联词抽取方法进行评测: 1) 基于依赖关系模板的关联词抽取方法; 2) 基于顺序模板的关联词抽取方法; 3) 基于上下文分析的关联词抽取方法[5]. 第一种方法和第三种方法在第 2.2 节进行了详尽的描述, 第二种方法类似于第一种方法, 只是将关联词局限于 Target 的前后两个词, 而不是上下两层依赖关系. 表 3 是利用上述三类关联词分别回答 TREC 2006 的定义类问题的评测结果. 从测试数据中不难看出, 基于依赖关系的关联词抽取方法明显好于其他两种方法, 比上下文分析法和基于顺序模板方法的成绩分别提高了 0.008 和 0.006.

表 3　关联词抽取方法的对比

Table 3　The comparison of related word extractions

关联词抽取方法	F 成绩
上下文分析法	0.303
基于位置顺序的关联词抽取方法	0.305
基于依赖关系的关联词抽取方法	0.311

3.3　MIRA 算法

3.3.1　特征向量

在上述两个实验中, 分别评价了定义句模板和关联词向量这两类特征的有效性. 本节将进一步讨论如何联合使用这两类特征, 并运用 DK-MIRA 估计其权重. 对于任意一个候选定义句 s, 我们首先计算其与关联词向量之间的相似性作为第一类特征. 对于第二类特征, 即定义句模板则采用常用的阈值过滤法, 我们提取频率超过 10 次的模板作为特征向量. 假设这些模板 $P=\{p_1,p_2,\cdots,p_k\}$, 则对应候选句 s 的第二类特征定义如下:

$$\psi(s,P)=\psi(s,p_j)_{j=1}^{k} \tag{12}$$

其中, 第 j 个模板 $\psi(s,p_j)$ 定义如下:

$$\psi(s,p_j)=\begin{cases}1, & \text{若 } s \text{ 蕴含模板 } p_j\\ 0, & \text{否则}\end{cases} \tag{13}$$

3.3.2　MIRA, K-MIRA 与 DK-MIRA

在学习各项特征权重之前, 首先对参数 k 进行估计. 我们设计实验使 k 取值从 1 到 10, 然后提取最佳观察值. 当 k 取 2 和 3 时, 本文的 K-MIRA 与 DK-MIRA 分别得到最佳成绩. 从实验结果 (见表 4 所示) 中我们可以看出, 依赖关系模板特征明显好于顺序模板特征, 对应在 MIRA, K-MIRA 和 DK-MIRA 模型中分别提高了 0.004, 0.003 和 0.005. 另外, 当在模板特征基础上加入基于依赖关系抽取的关联词向量时, 成绩进一步分别提高了 0.005, 0.011 和 0.018. 从分类器模型来看, 较之传统的 MIRA 算法, K-MIRA 和 DK-MIRA 有了较为明显的改进. 同时我们还可以看出, 同 K-MIRA 相比, DK-MIRA 在未加入关键词向量之前, 分别在顺序模板和依赖关系模板提高了 0.001 和 0.003. 而当依赖关

系模板加入关键词向量这个特征时, 其成绩则提高了 0.010, 效果显著. 从这一点上来说, 训练特征越有效, 则 DK-MIRA 算法的性能越能得到加强.

3.3.3 与其他系统对比

与 DK-MIRA 一并评测的系统包括: Soft Pattern (SP)[5], Human Interest Model (HIM)[22] 和 FDUQA[23]. 其中 SP, HIM 的数据均由文献 [22] 提供, FDQUA 是我们之前开发的系统 (TREC 2007 年官方排名第一[2]). 为了评测的客观性, 均采用与文献 [22] 相同的配置和评测工具 "Pourpre v1.0c". 从实验结果 (见表 5 所示) 中可以看出, 同 SP 相比, DK-MIRA 取得明显改进, F 成绩提高了 10.5 %; 而较之 HIM 与 FUDQA, DK-MIRA 仍然保持优势, 分别提高了 4.6 % 和 2.0 %.

表 4 MIRA, K-MIRA 和 DK-MIRA 之间的比较

Table 4 Comparison of MIRA, K-MIRA, and DK-MIRA

特征	MIRA	K-MIRA	DK-MIRA
顺序模板	0.306	0.312	0.313
依赖模板	0.310	0.315	0.318
依赖模板 + 关联词	0.315	0.326	0.336

表 5 与其他优秀系统对比数据

Table 5 Comparison of other advanced systems

系统	F 成绩
SP[22]	0.287
HIM[22]	0.303
FUDQA[23]	0.311
DK-MIRA	0.317

4 结论

候选句子排序作为定义类问题回答的一个核心任务, 通常分两步完成: 首先充分抽取有效特征, 然后将这些特征应用到排序算法或分类器. 本文的研究工作主要基于依赖关系的知识抽取, 包括定义句模板和关联词向量. 从实验数据可以看出, 较之仅仅利用位置顺序抽取的知识, 这种基于依赖关系的特征更加有效. 然后我们将这些特征应用到在线算法 MIRA, 从而实现对候选句子的排序. 文献 [24] 最早提出 MIRA 算法, 并从数学上系统地论证了 MIRA 的收敛性和最大错误分类边界值. 随后 McDonald 与 Taskar 等对 MIRA 算法进行了扩展, 用以实现对结构化数据分类. 本文在改进 MIRA 算法基础上, 提出 K-best MIRA (K-MIRA) 与动态 K-best MIRA (DK-MIRA) 算法. 这种新型的 MIRA 算法能够根据学习进程自动调整约束条件, 从而提高算法的收敛速度与性能. 同传统的 MIRA 相比, K-MIRA 与 DK-MIRA 均有较大的改进, 其中 DK-MIRA 更为明显.

致谢

感谢 Udine 大学的 Luca Di Gaspero 先生提供二次规划问题求解算法 (Goldfarb 算法).

References

1 Voorhees E M. Overview of the TREC 2003 question answering track. In: Proceedings of the 15th Text Retrieval Conference. Gaithersburg, USA: KFUPM ePrints, 2003. 1−13

2 Dang H T, Kelly D, Lin J. Overview of the TREC 2007 question answering track. In: Proceedings of the 15th Text Retrieval Conference. Gaithersburg, USA: KFUPM ePrints, 2007. 1−17

3 Cao J K, Wu L D, Huang X J, Zhou Y Q, Liu F. Using multiple combined ranker for answering definitional questions. In: Proceedings of the 14th Asia Information Retrieval Symposium. Harbin, China: Springer, 2008. 448−453

4 Cui H, Kan M Y, Chua T S. Unsupervised learning of soft patterns for generating definitions from online news. In: Proceedings of the 13th International World Wide Web Conference. New York, USA: ACM, 2004. 90−99

5 Cui H, Kan M Y, Chua T S. Generic soft pattern models for definitional question answering. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Salvador, Brazil: ACM, 2005. 384−391

6 Atallah M J. *Algorithms and Theory of Computation Handbook*. New York: CRC Press, 1999. 10−17

7 Crammer K, Singer Y. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 2003, **3**: 951−991

8 Collins M. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In: Processings of the 2002 Conference on Empirical Methods in Natural Language Processing. Philadelphia, USA: ACL, 2002. 1−8

9 Zhao Yu, Xi Hong-Sheng, Wang Zi-Lei, Yang Jian. Online training of SVM for multiuser detection. *Journal of System Simulation*, 2006, **18**(1): 50−53
(赵宇, 奚宏生, 王子磊, 杨坚. 基于在线 SVM 的多用户检测算法及仿真. 系统仿真学报, 2006, **18**(1): 50−53)

10 Dai Liu-Ling, Huang He-Yan, Chen Zhao-Xiong. An on-line learning algorithm for text categorization with support vector machines. *Journal of Chinese Information Processing*, 2005, **19**(5): 11−16
(代六玲, 黄河燕, 陈肇雄. 一种文本分类的在线 SVM 学习算法. 中文信息学报, 2005, **19**(5): 11−16)

11 Gu Ping, Zhu Qing-Sheng. A parallelizable Bayesian online learning algorithm. *Computer Science*, 2006, **33**(4): 159−161
(古平, 朱庆生. 一种可并行的贝叶斯集合在线学习算法. 计算机科学, 2006, **33**(4): 159−161)

12 Crammer K, Dekel O, Keshet J, Shalev-Shwartz S, Singer Y. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 2006, **7**: 551−585

13 McDonald R, Crammer K, Pereira F. Online large-margin training of dependency parsers. In: Proceedings of the 43rd Annual Meeting on the Association for Computational Linguistics. Ann Arbor, USA: ACL, 2005. 91−98

14 Lin D K. Principar: an efficient, broad-coverage, principle-based parse. In: Proceedings of the 15th International Conference on Computational Linguistics. Kyoto, Japan: ACL, 1994. 482−488

15 Han K S, Song Y I, Rim H C. Probabilistic model for definitional question answering. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, USA: ACM, 2006. 212−219

16 Dempster A P, Laird N M, Rubin D B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 1977, **39**(1): 1−38

17 Radev D R, Jing H Y, Stys M, Tam D. Centroid based summarization of multiple documents. *Information Processing and Management*, 2004, **40**(6): 919−938

18 McDonald R. Discriminative sentence compression with soft syntactic evidence. In: Proceedings of European Chapter of the Association for Computational Linguistics. Trento, Italy: ACL, 2006. 297−304

19 Taskar B, Guestrin C, Koller D. Max-margin Markov networks. In: Processings of the 17th Annual Conference on Neural Information Processing Systems. Vancouver, Canada: MIT, 2004. 25−32

20 Taskar B, Klein D, Collins M, Koller D, Manning C. Max-margin parsing. In: Processings of the Conference on Empirical Methods in Natural Language Processing. Barcelona, Spain: ACL, 2004. 1−8

21 McDonald R, Hannan K, Neylon T, Wells M, Reynar J. Structured models for fine-to-coarse sentiment analysis. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. Prague , Czech Republic: ACL, 2007. 432−439

22 Kor K W, Chua T S. Interesting nuggets and their impact on definitional question answering. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM, 2007. 335−342

23 Qiu X P, Li B, Shen C, Wu L D, Huang X J, Zhou Y Q. FDUQA on TREC 2007 QA track. In: Proceedings of the 16th Text Retrieval Conference. Gaithersburg, USA: NIST, 2007. 1−8

24 Crammer K, Singer Y. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2001, **2**: 265−292

曹均阔 复旦大学计算机学院博士研究生. 2003 年获中国海洋大学计算机应用专业硕士学位. 主要研究方向为自然语言处理. 本文通信作者.
E-mail: junkuocao@gmail.com
(**CAO Jun-Kuo** Ph. D. candidate at Fudan University. He received his master degree from Ocean University of China in 2003. His main research interest is natural language processing. Corresponding author of this paper.)

黄萱菁 复旦大学计算机学院教授. 1998 年获复旦大学计算机系博士学位. 主要研究方向为自然语言处理.
E-mail: xjhuang@fudan.edu.cn
(**HUANG Xuan-Jing** Professor at Fudan University. She received her Ph. D. degree in the Department of Computer Science, Fudan University in 1998. Her main research interest is natural language processing.)