

一种基于云模型的面向查询多文档自动文摘方法

陈劲光¹, 何婷婷²

1. 湖州师范学院教师教育学院, 湖州, 中国, 313000

2. 华中师范大学计算机学院, 武汉, 中国, 430079

1. c.jg2003@hutc.zj.cn, 2. tthe@mail.ccnu.edu.cn

【摘要】本文提出了一种利用云模型生成面向查询多文档自动文摘的新方法。针对面向查询多文档自动文摘的特点,在文摘的多个阶段定义相应的云,同时考虑文摘生成过程的随机性和模糊性,提高自动文摘的效果。在DUC(Document Understanding Conference, 文本理解会议)2005至2007以及TAC(Text Analysis Conference, 文本分析会议)2008-2009等大规模公开测试语料上进行的实验表明,基于云模型的方法可以有效改善生成文摘的质量,将相对简单的原型系统改造为效果较好的文摘系统。在刚刚结束的TAC 2010评测中,本文的方法取得了良好的成绩。

【关键词】云模型; 自动文摘; 不确定性

Generating Query-focused Multi-document Summarization Using Cloud Model

Jinguang Chen¹, Tingting He²

1. School of Teacher Education, Huzhou Teachers College, Huzhou 313000, China

2. College of Computer Science, Huazhong Normal University, Wuhan 430079, China

1. c.jg2003@hutc.zj.cn, 2. tthe@mail.ccnu.edu.cn

Abstract: This paper presents a novel method of improving effectiveness of query-focused multi-document summarization by handling uncertainties with cloud model. Since clouds are defined corresponding to the key phases of summarization, contributions of words and sentences are obtained by taking into account both fuzziness and randomness of their basic distribution. Experimental results show that our method achieved quite promising results on the DUC2005, DUC2006, DUC2007, TAC2008, TAC2009 corpuses. Our method also achieved very good results in participating TAC 2010.

Keywords: cloud model; query-focused multi-document summarization; uncertainty

1 引言

Internet 的飞速发展极大方便了人们对信息的获取和使用,但同时也带来了信息过载的问题。在这些海量信息中快速、准确地找到所需要的信息变得越来越困难。面向查询的多文档自动文摘将查询返回的文档集合的内容提炼成包含与查询相关的主题、满足个性化需求的摘要,它能够显著提高信息获取和利用的效率。

目前,自动文摘系统大多采用直接从文档集合中抽取完整句子的方法生成文摘,从而避免生成不完整或者不符合语法的句子。基于抽取的方法主要有:基于高频

词的方法^[1],基于图的方法^[2],基于主题的方法^[3],以及基于语义的方法等。这些现有方法基本可以概括为:找到某个粒度下的文摘单元(例如词,短语,块,要素,句子或段落等)的某种随机分布,利用统计、更复杂的语言模型或者其他方法来解析这些分布,并据此生成摘要。然而,虽然各种随机分布的随机性被普遍关注,这些分布的另外一个重要方面,即不确定性问题却很少被考虑。

数学上一般采用熵、模糊集、粗糙集来研究不确定性问题。在文摘领域,目前已有关于将熵和模糊集用于生成自动文摘的研究。利用熵^[4,5]生成文摘的方法主要是最大熵方法,但该方法一般需要训练语料,并且只考虑了文摘单元在训练语料中分布的不确定性,而没有考虑当前文档集合分布的不确定性。利用模糊集生成摘要的方法大致可以分为两类:利用模糊性规则的方法^[6,7],以

本文获国家自然科学基金重大研究计划培育项目“汉语语义知识获取与语义计算模型研究”(项目编号:0920005),教育部人文社会科学研究青年项目“多媒体阅读中检索行为的发生机制与检索策略研究”(项目编号:13YJCZH013)资助。

及语义模糊共指链^[8]的方法。但是，应用模糊集的方法往往将随机分布的模糊性和随机性孤立开来，对随机分布的统计结果一般被用于决定隶属度函数的隶属度，从而实现对模糊问题的精确求解，随机性和模糊性之间的内在联系却很少被注意。

李德毅院士提出了云模型^[9,10]理论来实现自然语言表达的定性概念与定量值的相互转换。该理论融合熵、模糊集等不确定性理论的思想，以基本语言值为突破口，重视随机性和模糊性的内在联系，可以用来体现语言思考中的软推理能力。云模型已经在知识表示^[11]、关联规则挖掘^[12]、时间序列预测^[13]、自动控制^[14]等多个领域有着广泛应用。文献^[15]提出了一种利用云模型进行自动文摘评价的方法，但该方法实验语料规模偏小。

2 基于云模型的面向查询多文档自动摘要

2.1 文摘原型系统

原型系统采用一种多特征融合的方法，同时考虑句子的查询相关性和句子在文档集合中的重要程度两方面的因素，抽取句子，生成摘要。句子和查询越相关，在文档集合中的地位越重要，越有可能被选择为文摘句。这是面向查询的多文档自动文摘常用的方法。

首先，计算句子的查询独立特征(Query-independent, QI)，即不考虑查询条件的情况下，句子在文档集合中的重要度。

本文采用向量空间模型^[16] (Vector Space Model, VSM) 计算句子之间的相似度。对于给定文档集合，将每个句子表示为 m 维的向量 $(w_{i1}, w_{i2}, \dots, w_{im})$ ，其中 m 是文档集合的词种数，向量空间中的每一维对应着词表中的一个词。向量中每个元素的权重用该元素所在的维所对应的词语的 TF-ISF^[17] 得分来表示，即：

$$w_{ik} = \text{TF} \cdot \text{ISF} \quad (1)$$

句子之间相似度可以用向量之间的内积计算：

$$\text{Sim}(S_i, S_j) = \frac{\sum_{k=1}^m w_{ik} w_{jk}}{\sqrt{(\sum_{k=1}^m w_{ik})^2 (\sum_{k=1}^m w_{jk})^2}} \quad (2)$$

句子的查询独立特征值可以用该句子与所有句子的相似度之和来表示：

$$QI(S) = \sum_{i=1}^N \text{Sim}(S, S_i) \quad (3)$$

其次，计算每个句子的查询相关特征 (Query-focused, QF)，即句子与查询条件之间的相关程度。

Hyperspace Analogue to Language (HAL)理论^[18]认为自然语言素材中词与词之间的关系提供了足够的语义信息。文献^[19]首次将这种方法应用到自动文摘领域。

本文沿用文献^[19]提出方法获取句子和查询之间的相关特征，但与文献^[19]不同的是，该文献在计算了词与查询相关特征后，采用复杂的语言模型系统计算句子与查询之间的相关度，而本文则采用相对简单的统计方法计算句子与查询之间的相关度。

首先利用 HAL 模型计算文档集合中的词语与查询词之间的关联度，该方法可以被形象的称为窗口共现的方法，利用词与查询词在一定窗口长度内的共现情况计算词语与查询词之间的相关程度，从而获取词语与查询词之间的语义关联信息。在一个长度为 K (文献^[19]中 $K=8$ ，本文沿用了这个参数值) 的窗口范围内观察文档集合中的词语 (w) 与查询词 (w') 的共现情况，然后将这个窗口在整个文档集合范围内移动，每次向前移动一个词语。统计词语与查询词在一定距离的共现情况，距离越小，共现次数越多，则说明该词语与查询词越相关。

设 $n(w, k, w')$ 代表 w 与 w' 在距离为 k 的共现次数， $W(k)=K-k+1$ 代表词语 w 与 w' 的共现强度。则词语与查询词的相关程度可表示为：

$$\text{HAL}(w'|w) = \sum_{k=1}^K W(k)n(w, k, w') \quad (4)$$

词语与查询条件的相关程度可表示为词语与每个查询词相关程度的累加和：

$$QF(w|Q) = \sum_{j=1}^J \text{HAL}(w|q_j) \quad (5)$$

文档集合中的句子的查询相关得分可表示为句子中每个词语与查询条件相关程度的累加和：

$$QF(S) = \sum_{i=1}^I QF(w_i|Q) / I = (\sum_{i=1}^I \sum_{j=1}^J \text{HAL}(w_i|q_j)) / I \quad (6)$$

句子的最终得分由查询独立得分与查询相关得分线性组合得到：

$$\text{Score}(S) = \sigma \cdot \frac{QF(S)}{\sum_{i=1}^N QF(S_i)} + (1-\sigma) \cdot \frac{QI(S)}{\sum_{i=1}^N QI(S_i)} \quad (7)$$

其中 σ 是调节两部分比例的调节参数。

2.2 查询独立云

针对 2.1 节的句子的查询独立特征，我们提出查询独立云，通过对查询独立过程中的不确定性的把握来改进原型系统中计算句子的查询独立得分的过程。

在原型系统中，一个句子如果与文档集合中所有句子相似度之和最高，就会获得最高的查询独立得分。然而，相似度之和最高并不意味着该句子就能很好的概括文档集合中所有的句子的内容。例如当一个文档集合包

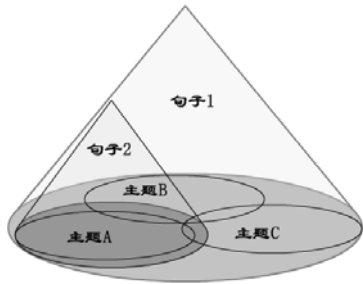


图1 自动文摘过程中的局部聚焦问题

含多个主题的时候，生成文摘过程中往往出现局部聚焦问题，即生成的文摘往往只概括其中一个主题的内容。在图1中，句子1能够概括文档集合所要表达的所有主题，而句子2仅能概括主题A的内容，句子1一般来说更适合被选做文摘句；但由于句子2是局部聚焦，不像句子1要概括很多方面的内容，有时更容易取得高分，从而被选为文摘句。

该问题可以通过引入云模型来解决。虽然传统的统计方法无法分辨句子1和句子2的优劣，但句子1和句子2从不确定性的角度看还是有差异的：句子1由于要概括所有的主题，势必和每个主题中的句子都比较相似，因此句子1和所有句子的相似度分布比较均匀，不确定性比较小；句子2由于只概括了主题A，仅仅与主题A中的句子比较相似，而与主题B和主题C中的句子基本不相似，因此句子2和所有句子的相似度分布比较不均匀，不确定性比较大。用云模型来描述，句子1和所有句子的相似度构成的云，相对于句子2和所有句子的相似度构成的云来说，具有更小的熵和超熵。

查询独立云的思想是：

将句子和文档集合中每个句子的相似度看成云滴，用这些云滴构成的云的不确定性来评价句子概括文档集合内容的的能力。一个句子所对应的云的期望越大，熵和超熵越小，就认为该句子越能代表文档集合的内容，查询独立得分也越高。

设文档集合中有N个句子，该集合中句子间的相似度可以用一个 $N \times N$ 的相似度矩阵 \tilde{M} 来表示。矩阵 \tilde{M} 的每一行表示一个句子与所有句子的相似度，可以将每行看成一片云，用来衡量该行所对应的句子概括文档集合内容的的能力。对于矩阵 \tilde{M} 的每一行，可以利用逆向云获取云的数字特征 Ex, En, He 。这样的N片云构成的云团可以用一个 $N \times 3$ 的矩阵 \tilde{A} 来表示：

$$\tilde{A} = [\vec{A}_1, \vec{A}_2, \vec{A}_3] = [Ex, En, He] = CG^{-1}(\tilde{M}) \quad (8)$$

将期望、熵、超熵按一定的权值线性组合，就可以用一个 $N \times 1$ 的列向量表示句子的查询独立得分：

$$QICS\text{Score} = \alpha_1 \cdot \frac{\overline{A}_1}{\max(A_1)} - \beta_1 \cdot \frac{\overline{A}_2}{\max(A_2)} - \gamma_1 \cdot \frac{\overline{A}_3}{\max(A_3)} \quad (9)$$

$$\alpha_1 + \beta_1 + \gamma_1 = 1, \alpha_1 > \beta_1 > \gamma_1 \geq 0$$

公式(9)通过每一列除以该列的最大值进行了归一化处理。 $\alpha_1, \beta_1, \gamma_1$ 是线性组合的权值参数，由于期望作用一般大于熵和超熵，而熵的作用一般大于超熵，因此 $\alpha_1 > \beta_1 > \gamma_1 \geq 0$ 。当 $\alpha_1 = 1, \beta_1 = \gamma_1 = 0$ 时，查询独立得分仅考虑了句子与所有句子之间相似度的期望值，其结果与原型系统一致。

2.3 查询相关云

针对2.1节的句子的查询相关特征，我们提出查询相关云，通过对查询相关过程中的不确定性的把握来改进原型系统中计算查询相关得分的过程。查询相关云由词语级查询相关云、句子级查询相关云两部分组成，这两块云分别处理词语、句子两个不同层面的查询相关云。

例如当查询条件为“毛泽东的故乡是哪里？”，可能有很多词语与“毛泽东”有关，或者与“故乡”有关，但只有同时与“毛泽东”和“故乡”两个词语都有关的词语如“韶山”才是准确的回答。

表1给出了TAC09中D0901A-A文档集合中的一个例子，该集合主题为“印巴冲突”，查询词有：“描述”、“努力”、“和平”、“印度”、“巴基斯坦”、“冲突”、“克什米尔”。文档集合中有5个词语与以上7个查询词查询相关得分的期望相同，它们是：“努力”、“会谈”、“信号”、“maharajah”、“停火”。在原型系统中，以上5个词语将得到相同的打分，而实际上“努力”、“会谈”分别与4个查询词有共现，“信号”、“maharajah”仅与3个查询词有共现，“停火”则仅与两个查询词共现。事实上，“信号”到“和平、努力”在含义上还有一段距离，“maharajah”指的是印度的地区领导者，很有可能只是在介绍这个地区情况的

	describe	effort	peace	indium	pakistan	conflict	kashmir	Ex	En	He	WQFCScore
effort	0	0	7	5	0	4	3	2.71	2.92	0.77	1.24
meet	0	8	0	0	8	2	1	2.71	3.79	0.87	1.06
signal	0	0	0	8	7	0	4	2.71	3.89	1.49	0.97
maharajah	0	0	0	8	6	0	5	2.71	3.89	1.70	0.95
ceasefire	0	0	0	0	14	0	5	2.71	4.86	2.15	0.71

表1 查询相关云的例子

句子中出现。“停火”理论上应当与“和平、进程”很有关系，但分析语料发现，2003年巴基斯坦方面开始了单方面停火，其后双方更多的开展的是一系列和平谈判，所以“停火”这个词语大多是在谈到“巴基斯坦方面单方面停火”时才会提到，因此该词语并不是一个与查询非常相关的词语。

词语级查询相关云的基本思想是：

将文档集合中的词语与各个查询词之间的相关度看作是云滴，用这些云滴构成的云来描述词语与查询条件之间的相关性。期望相同的情况下，一个词语和越多的查询词相关，它所对应的云的熵和超熵越小，该词语就被认为和查询条件越相关。

表1中，每一行中词语和查询词之间的相关度被看成云滴，这些云滴构成一片云，利用逆向云可以计算它们的数字特征，这些数字特征经过线性组合可以计算出每个词语的查询相关得分，在表1中可以看到这些计算结果。

设查询条件中有 m 个查询词，文档集合的词种数为 n ，利用原型系统中提到的窗口共现方法，我们可以获得类似于表1第1-8列结构的 $n \times m$ 的矩阵 \overline{HAL} 。将 \overline{HAL} 的每一行看成云，利用逆向云可以获得每一行的数字特征，各个词语对应的云组成的云团可以用一个 $n \times 3$ 的矩阵 \overline{B} 来表示：

$$\overline{B} = [B_1, B_2, B_3] = [\overline{Ex}, \overline{En}, \overline{He}] = CG^{-1}(\overline{HAL}) \quad (10)$$

词语与查询条件的相关程度可以表示为一个 $n \times 1$ 的列向量 $\overline{WQFCScore}$ ：

$$\overline{WQFCScore} = \alpha_1 \cdot \frac{\overline{B}_1}{\max(B_1)} - \beta_2 \cdot \frac{\overline{B}_2}{\max(B_2)} - \gamma_2 \cdot \frac{\overline{B}_3}{\max(B_3)} \quad (11)$$

$$\alpha_1 + \beta_2 + \gamma_2 = 1, \alpha_1 > \beta_2 > \gamma_2 \geq 0$$

可以看出， $\overline{WQFCScore}$ 的计算方式与 $\overline{QICScore}$ 非常相似。

在获取了每个词语与查询条件相关程度以后，需要给文档集合中的句子打分，原型系统中通过计算句子中各个词语与查询条件相关程度得分的期望计算句子的得分，这种做法同样可以通过引入云模型加以改进。

句子级查询相关云的基本思想为：

将句子中由前一阶段获得的词语与查询句之间的相关度看作是云滴，用这些云滴构成的云来描述句子与查询条件之间的相关性。期望相同的情况下，一个句子中有越多和查询句相关的词，它所对应的云的熵和超熵越小，该句子就被认为和查询条件越相关。

设句子 S_i 由 k 个词组成，则它可以表示为行向量 $\overline{S}_i = [w_1, w_2, \dots, w_k]$ ，其中 w_k 为该句子中第 k 个词语与查询条件的相关程度。将该向量中的每一个元素的值看

成云滴，则我们可以用逆向云计算这些云滴构成的云的数字特征：

$$\overline{C}_i = [Ex_i, En_i, He_i] = CG^{-1}(\overline{S}_i) \quad (12)$$

有多少个句子就有多少片云，这些云叠加在一起构成了云团，可以用一个 $N \times 3$ 的矩阵来表示：

$$\overline{C} = [\overline{Ex}, \overline{En}, \overline{He}] \quad (13)$$

我们可以用一个 $N \times 1$ 的列向量 $\overline{QFCScore}$ 表示句子的查询相关得分：

$$\overline{QFCScore} = \alpha_3 \cdot \frac{\overline{C}_1}{\max(C_1)} - \beta_3 \cdot \frac{\overline{C}_2}{\max(C_2)} - \gamma_3 \cdot \frac{\overline{C}_3}{\max(C_3)} \quad (14)$$

$$\alpha_3 + \beta_3 + \gamma_3 = 1, \alpha_3 > \beta_3 > \gamma_3 \geq 0$$

虽然 $\overline{QFCScore}$ 的计算过程与 $\overline{WQFCScore}$ 有很多相似之处，但需要注意的是，由于每个句子的词语数是不相同的，因此这些句子对应的云的云滴个数也是不相同的，这些云之所以能叠加在一起构成云团，是因为它们的数字特征的个数是相同的。

将查询独立云和查询相关云的计算结果按类似原型系统的方法进行线性叠加，就构成了云文摘模型 (Cloud Summarization, 简称 CloudSum)：

$$\overline{CloudSumScore} = \sigma \cdot \frac{\overline{QFCScore}}{\sum_{i=1}^N \overline{QFCScore}} + (1-\sigma) \cdot \frac{\overline{QICScore}}{\sum_{i=1}^N \overline{QICScore}} \quad (15)$$

将 $\overline{CloudSumScore}$ 排序，去冗余，即可以生成文摘。

3 实验效果评测

3.1 实验语料

本文采用了 DUC¹ 2005、2006、2007(简称 DUC05、DUC06、DUC07)的面向查询多文档自动文摘任务语料，以及 DUC 2007、TAC² 2008、TAC 2009 文摘任务 A 集合 (简称 DUC07-U、TAC08-U、TAC09-U) 的语料作为实验语料。表2列出了实验语料基本情况。

3.2 实验过程

为了专注于研究云模型在自动文摘中的作用，以及便于研究者重复我们的实验，在整个实验过程中，除了采用云方法，我们在实验的其他环节尽可能采取比较简单、且开源的方法进行实验。实验的预处理阶段主要包括三个部分的内容：句子切分、去停用词、词干化。DUC 为参赛队伍提供了开源的句子切分工具，breaksent-multi.pl³，我们采用该工具将文档集合中的句子以及查询条件切分成句子。在去停用词后，我们采用

¹ <http://duc.nist.gov/>

² <http://www.nist.gov/tac/>

³ www-nlpir.nist.gov/projects/duc/duc2007/tasks.html

斯坦福大学开发的词干化工具 Morphology⁴对词语进行词干化, Morphology 是基于由 Minnen 等人^[19]提出的采用有限状态自动机的原理获取英文单词的原型形态的方法开发的一种开源词干化工具。Morphology 的词干化效果一般来说优于传统的基于规则的词干化工具, 词干化后的词语只是转化为原始形态, 但仍然是完整的单词形式。例如前面表 1 提到的“India”就被词干化为“Indium”。在词干化以后, 所有单词被转换为小写的形式, 包含连接符“-”的词语被转换切分为两部分, 其他的标点符号被去掉。另外, 长度少于 3 个字母的词语一般来说包含的有效信息较少, 因而也被从词表中去掉。

句子经过打分以后, 后期处理过程主要是去冗余。我们采用一种改进的 MMR (Maximal Marginal Relevance Maximal Marginal Relevance)^[20]方法去冗余。其基本思想是: 每选择好一个文摘句, 就将该文摘句对剩下的候选文摘句的影响去除掉; 循环的进行这样的操作, 相当于去除了即将选择的文摘句与所有已经选择好的文摘句的相关信息。

$$S = \arg \max_{S_i \in R-F} [Score(S_i) - Score(S_L) \cdot sim(S_i, S_L)] \quad (16)$$

其中 R 是所有句子的集合, 而 F 是所有已经选择好的文摘句的集合, 因而 S_i 表示候选文摘句; S_L 表示最近选取的文摘句。公式 16 所表示的过程是循环进行的, 不断选择文摘句, 直到达到指定的文摘长度。

本文的方法有大量的参数需要训练, 但由于以上参数都有约束条件, 我们发现只需要使用简化的训练过程就取得了满意的实验效果。详细的训练过程, 这里由于篇幅的限制不能展开。具体训练过程中, 我们采用 DUC06 的语料作为训练语料训练 DUC05、DUC07 的参数, 采用 TAC08-U 的语料作为训练语料训练 DUC07 和 TAC09-U 的参数。最终, 我们采用

$$\begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{bmatrix} = \begin{bmatrix} 0.8 & 0.15 & 0.05 \\ 0.8 & 0.15 & 0.05 \\ 0.8 & 0.15 & 0.05 \end{bmatrix} \quad \sigma = 0.1$$

作为 DUC05、DUC07 的参数,

$$\begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{bmatrix} = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.7 & 0.2 & 0.1 \\ 0.7 & 0.2 & 0.1 \end{bmatrix} \quad \sigma = 0.2$$

作为 DUC07 和 TAC09-U 的参数。

3.3 实验结果

表 3 显示了 CloudSum 方法与当年参加评测得分最高的机器系统的 ROUGE-2 得分。从表 4 中我们可以看出, CloudSum 在 DUC05、DUC07-U 中位列所有机器系统第 3 位, 在 DUC07、TAC09-U 中也取得了接近前

	DUC05	DUC06	DUC07	DUC07-U	TAC08-U	TAC09-U
文档集合数	50	50	45	10	48	44
总文档数	1593	1250	1125	100	480	440
文摘长度	250	250	250	100	100	100
文档集合平均大小(kb)	114	66	50	32	29	31

表 2. 实验语料基本情况.

	DUC05	DUC07	DUC07-U	TAC09-U
Human	0.089	Human 0.130	M1 0.112	M1 0.122
M1	0.073	M1 0.124	Human 0.105	M2 0.121
M2	0.072	M2 0.120	M2 0.099	Human 0.111
CloudSum	0.070	M3 0.119	CloudSum 0.097	M3 0.109
M3	0.070	M4 0.118	M3 0.096	M4 0.106
M4	0.070	M5 0.112	M4 0.094	M5 0.105
M5	0.069	CloudSum 0.106	M5 0.094	CloudSum 0.104
AvgM	0.058	AvgM 0.095	AvgM 0.074	AvgM 0.085

表 3 CloudSum 的 ROUGE-2 得分, 其中 Human 是得分最低的人工系统, M1, M2, M3, M4, M5 得分最高的前五个自动文摘系统, AvgM 是机器系统的平均得分。

	DUC05	DUC07	DUC07-U	TAC09-U
原型系统	0.067	0.099	0.081	0.099
CloudSum	0.070	0.106	0.097	0.104

表 4 原型系统与 CloudSum 的 ROUGE-2 得分比较

2010 年我们采用 CloudSum 参加 TAC 2010 有导文摘国际比赛, 为了将有导的任务与面向查询自动文摘任务相联系, 我们仅将组委会给定的类别提示信息作为查询条件, 其他方面都与之前的实验保持一致。我们提交了两个系统, ID 分别为 6、23 号, CloudSum 编号为 23 号 (图 3.16)。

在 TAC 2010 评测数据集 A (其任务与面向查询的多文档自动文摘相似) 上, 基于云模型的文摘其 ROUGE-2、ROUGE-SU4、Basic Elements、人工综合评测指标 Average Overall Responsiveness 四个自动评测指标分别取得了 43 个参评系统中排名第 3、第 2、第 2、第 3 的成绩。图 3.17 显示了各项评测结果。

⁴ <http://nlp.stanford.edu/software/tagger.shtml>

D	0.12862	(95%-conf.int. 0.11087 - 0.14681)
H	0.12841	(95%-conf.int. 0.11058 - 0.14771)
F	0.12556	(95%-conf.int. 0.10931 - 0.14155)
C	0.11695	(95%-conf.int. 0.09822 - 0.13447)
B	0.11313	(95%-conf.int. 0.09197 - 0.13365)
G	0.10558	(95%-conf.int. 0.08716 - 0.12527)
E	0.10493	(95%-conf.int. 0.08817 - 0.12344)
A	0.09623	(95%-conf.int. 0.08074 - 0.11167)
22	0.09620	(95%-conf.int. 0.08922 - 0.10314)
18	0.09451	(95%-conf.int. 0.08769 - 0.10093)
23	0.09418	(95%-conf.int. 0.08669 - 0.10163)
36	0.09213	(95%-conf.int. 0.08539 - 0.09854)
24	0.09202	(95%-conf.int. 0.08592 - 0.09819)
31	0.08743	(95%-conf.int. 0.08126 - 0.09293)
6	0.08728	(95%-conf.int. 0.08042 - 0.09409)
34	0.08624	(95%-conf.int. 0.08027 - 0.09215)
13	0.08554	(95%-conf.int. 0.07956 - 0.09160)
10	0.08488	(95%-conf.int. 0.07859 - 0.09144)

ROUGE-2

H	0.16294	(95%-conf.int. 0.14894 - 0.17821)
F	0.16212	(95%-conf.int. 0.14759 - 0.17574)
D	0.16200	(95%-conf.int. 0.14865 - 0.17607)
C	0.15289	(95%-conf.int. 0.13691 - 0.16798)
E	0.14637	(95%-conf.int. 0.13212 - 0.16064)
G	0.14513	(95%-conf.int. 0.13070 - 0.16034)
B	0.14396	(95%-conf.int. 0.12833 - 0.16071)
A	0.13805	(95%-conf.int. 0.12685 - 0.14972)
22	0.13049	(95%-conf.int. 0.12489 - 0.13585)
23	0.12978	(95%-conf.int. 0.12332 - 0.13614)
24	0.12834	(95%-conf.int. 0.12267 - 0.13379)
18	0.12439	(95%-conf.int. 0.11889 - 0.12950)
34	0.12284	(95%-conf.int. 0.11759 - 0.12829)
36	0.12278	(95%-conf.int. 0.11748 - 0.12797)
13	0.12258	(95%-conf.int. 0.11783 - 0.12749)
9	0.12170	(95%-conf.int. 0.11685 - 0.12663)
3	0.12096	(95%-conf.int. 0.11643 - 0.12535)
16	0.12084	(95%-conf.int. 0.11615 - 0.12533)

ROUGE-SU4

F	0.09114	(95%-conf.int. 0.07800 - 0.10449)
H	0.08690	(95%-conf.int. 0.06710 - 0.10748)
D	0.08449	(95%-conf.int. 0.06727 - 0.10376)
C	0.07727	(95%-conf.int. 0.06119 - 0.09319)
G	0.07442	(95%-conf.int. 0.05802 - 0.09326)
B	0.07382	(95%-conf.int. 0.05695 - 0.09199)
A	0.06749	(95%-conf.int. 0.05461 - 0.08337)
E	0.06696	(95%-conf.int. 0.05216 - 0.08341)
22	0.05973	(95%-conf.int. 0.05432 - 0.06615)
23	0.05828	(95%-conf.int. 0.05225 - 0.06428)
18	0.05772	(95%-conf.int. 0.05172 - 0.06412)
13	0.05576	(95%-conf.int. 0.05052 - 0.06100)
16	0.05523	(95%-conf.int. 0.04995 - 0.06078)
24	0.05361	(95%-conf.int. 0.04852 - 0.05926)
3	0.05272	(95%-conf.int. 0.04799 - 0.05795)
6	0.05240	(95%-conf.int. 0.04709 - 0.05733)
36	0.05235	(95%-conf.int. 0.04685 - 0.05785)
31	0.05116	(95%-conf.int. 0.04622 - 0.05623)

Basic-Element

1. peerID	2. average modified (pyramid) score	3. average numSCUs	4. average numrepetitions	5. macroaverage modified score with 3 models	6. average linguistic quality	7. average overall responsiveness
16	0.399	5.739	1.326	0.392	3.457	3.174
22	0.425	5.957	1.022	0.418	3.109	3.13
23	0.394	5.63	0.891	0.386	2.717	3.087
24	0.414	5.804	1.065	0.406	3.217	3.065
13	0.414	5.935	1.261	0.407	3.37	3.022
21	0.398	5.587	0.891	0.392	2.978	3.022
36	0.397	5.674	0.935	0.391	3.304	3.022
10	0.386	5.63	0.674	0.378	3.283	3.022
34	0.383	5.457	0.978	0.377	3.239	3.022
3	0.399	5.5	1.413	0.393	3.043	2.978

人工评测

图 2. 1 TAC 2010 评测数据集 A (其任务与面向查询的多文档自动文摘相似) 上, 其 ROUGE-2、ROUGE-SU4、Basic Elements、人工评测 Average Overall Responsiveness 分别取得了 43 个参评系统中排名第 3、第 2、第 2、第 3 的成绩

说明: A 到 H 为人工摘要, 1 到 43 为机器摘要 (仅列前十)

4. 结论

本文提出了一种利用云模型理论直接生成文摘的方法, 云模型被应用于文摘生成过程中的几个关键阶段。大规模公开评测的语料上进行的实验表明了本文方法的有效性。鉴于不确定性在自然语言处理中存在的普遍性, 以及云模型理论在数据挖掘等相关领域的成功, 本文提出的方法将有可能被应用于自动文摘以外的自然语言处理的其他领域。

参考文献

[1] K. Toutanova, C. Brockett, M. Gamon, J. Jagarlamundi, Hi. Suzuki, and L. Vanderwende, "The Pythy Summarization System: Microsoft Research at DUC 2007", In Proc. Document Understanding Conference, 2007.

[2] X.J. Wan and J.W. Yang, "Improved affinity graph based multi-document summarization", In Proc. HLT-NAACL, pp. 181-184, 2006.

[3] A. Haghighi and L. Vanderwende. "Exploring content models for multi-document summarization", In Proc. NAACL-HLT, pp. 362-370, 2009

[4] L. Ferrier, *A Maximum Entropy Approach to Text Sum-*

marization. School of Artificial Intelligence, Division of Informatics, University of Edinburgh, 2001.

[5] G. Ravindra, N. Balakrishnan, and K.R. Ramakrishnan. "Multi-Document Automatic Text Summarization Using Entropy Estimates", In Proc. SOFSEM, pp. 289-300, 2004.

[6] F.R. Isfahani, F. Kyoomarsi, H. Khosravi, E. Eslami, A. Tajodin, and P.K. Dehkordy, "Application of Fuzzy Logic in the Improvement of Text Summarization", In Proc. IADIS International Conference Informatics, pp. 347-352, 2008.

[7] M.S. Binwahlan, N. Salim, and L. Suanmali. "Fuzzy Swarm Based Text Summarization", *Journal of Computer Science*, vol. 5, no. 5, pp. 338-346, 2009.

[8] R. Witte and S. Bergler. 2003. "Fuzzy Coreference Resolution for Summarization", In Proc. International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS), pp. 43-50, 2003.

[9] D.Y. Li, X. Shi, and M.M. Gupta. "Soft Inference Mechanism Based on Cloud Models", In Proc. the 1st International Workshop on Logic Programming and Soft Computing: Theory and Applications (LPSC), pp. 38-62, 1996.

[10] 李德毅, 杜鹞, *不确定性人工智能*, 国防工业出版社, 2005 年第 1 版。

- [11] 邸凯昌, 李德毅, “云理论及其在空间数据发掘和知识发展中的应用”, *中国图象图形学报:A 辑*, vol. 4, no. 11, pp. 930~935, 1999.
- [12] 杜鹤, 宋自林, 李德毅, “基于云模型的关联规则挖掘方法”, *解放军理工大学学报(自然科学版)*, vol. 1, no. 1, pp. 29-34, 2000.
- [13] 蒋嵘, 李德毅, “基于形态表示的时间序列相似性搜索”, *计算机研究与发展*, vol. 37, no. 5, pp. 601-608, 2000.
- [14] D.Y. Li, H. Chen, J.H. Fan, C.Z. Shen, “A Novel Qualitative Control Method to Inverted Pendulum Systems”, In Proc. the 14th International Federation of Automatic Control World Congress, 1999.
- [15] H. Long, Z.H. He, S.Q. Li, J. You, “Automated Summarization Evaluation Based on Clouds Model”, In Proc. China Information Retrieval Conference (CCIR 2009), pp. 9-16, 2009.
- [16] G. Salton, A. Wong, and C.S. Yang, “A Vector Space Model for Automatic Indexing”, In *Communications of the ACM*, vol. 18, no. 11, pp. 613-620, 1975.
- [17] J.L. Neto, A.D. Santos, C.A.A. Kaestner, and A.A. Freitas. “Document clustering and text summarization”. In Proc. 4th Int. Conf. Practical Applications of Knowledge Discovery and Data Mining, pp. 41-55, 2000.
- [18] K. Lund and C. Burgess, “Producing high-dimensional semantic spaces from lexical co-occurrence”, *Behavior Research Methods, Instrumentation, and Computers*, vol. 28, pp.203-208, 1996.
- [19] J. Jagarlamudi, P. Pingali, and V. Varma, “A Relevance-Based Language Modeling Approach to DUC 2005”, In Proc. Document Understanding Conference, 2005.
- [20] G. Minnen, J. Carroll, and D. Pearce, “Applied morphological processing of English”. *Natural Language Engineering*, vol. 7, no. 3, pp. 207-223, 2001.
- [21] J.G. Carbonell and J. Goldstein, “The use of MMR, diversity-based re-ranking for reordering documents and producing summaries”, In Proc. SIGIR, pp. 335-336, 1998.
- [22] C.Y. Lin and E. Hovy. “Automatic evaluation of summaries using n-gram co-occurrence statistics”. In Proc. NLT-NAACL, pp. 71-78, 2003.