基于 FCA 的协作信息检索框架

李云　盛艳　沈岑诚　栾鸾
(扬州大学 信息工程学院，江苏扬州 225009)
E-mail: shengyan1985@gmail.com

摘要: 在协作信息检索中，针对多个子文本数据库，利用渐进式构格算法构建各个子形式概念格。然后分别利用形式概念相似度的衡量方法对已有形式概念进行匹配，在找到临时概念集之后采用合并的方式，获得新的概念。再对新的概念进行相似度的匹配，最终获得满足用户需求的结果集合。整个框架便于在分布式环境中部署实施，并且在匹配查询关键词时使用了不精确的方式，更好的体现了人性化需求。
关键词: FCA，协作，信息检索

Collaborative Information Retrieval Framework Based on FCA

Li Yun　Sheng Yan　Shen Cencheng Luan Luan
(Institute of Information Engineering，Yangzhou University，Jiansu Yangzhou 225009)
E-mail: shengyan1985@gmail.com

Abstract: In Collaborative Information Retrieval, we construct formal concept lattices for a number of sub-text databases by the progressive lattice-built algorithm. Then, respectively, we obtained the new formal concepts by merging the temporary formal concepts which are from the different lattices by measuring similarity between formal concepts and next, matching the new concepts to return the results which are to meet the user needs ultimately. The whole framework is easy to deploy in a distributed environment and match the query words in an imprecise way. It reflects the needs of humanity better.
Key word: FCA, collaborative, information retrieval

1 引 言

Web 信息与日俱增，其种类结构也是丰富多样，尤其文本信息更是巨大，如今信息检索领域已成为各位专家学者研究的热门领域。国内外针对信息检索领域中提出了很多方法及理论[1]，其中，使用形式概念分析理论来进行信息检索也有很多深入的研究，如，使用 FCA 构建表示信息模型以更确切的表示原始检索模型，并使用构建出来的概念格进行信息检索，基于抽象具体程度的不同进行检索可获得独特层面上的检索结果。另外，在分布式环境下，使用 FCA 理论和技术进行协作信息检索，这可大大提高检索时间和效率。

为了能在大数据集中更快速、更有效的检索出更符合用户需求的结果，使用基于语义的一系列策略。在协作信息检索中，针对多个子文本数据库，利用渐进式构格算法构建各个子形式概念格，然后分别利用形式概念相似度的衡量方法对已有形式概念进行匹配，在找到临时概念集之后采用合并的方式，获得新的概念，再对新的概念进行相似度的匹配，最终获得满足用户需求的结果集合。整个系统充分体现了协作的思想，便于在分布式环境中部署实施，并且在匹配查询关键词时使用了不精确的方式，从结构和语义两个层次上进行衡量，更好的体现了人性化需求。

2 Formal Concept Analysis

形式概念分析(FCA)[2]是一种建立在数论基础上的数据分析技术，其基本内容是形式背景(Formal Context)、形式概念(Formal Concept)以及形式概念之间的关系。形式背景(Formal Context)被定义为一个三元组 K=(G，M，I)，其中 G 和 M 分别是对象(object)集合和特征(属性 attribute)集合，而 I 是 G 和 M 之间的二元关系，即 I⊆G×M，gIm。在形式背景 K 中，在 G 的幂集和 M 的幂集之间可以定义如下的两个映射函数 f 和 g：

$$A' = \{m \in D | gRm, \forall g \in A\} \quad (1)$$

$$B' = \{g \in O | gRm, \forall m \in B\} \quad (2)$$

它们也称为 U 和 A 之间的 Galois 连接。来自 P(G)×P(M)的二元组(A， B)如果满足两个条件：A=g(B)和 B=f(A)或 A=B'和 B=A'，则称(A， B)为形式背景 K 的一个形式概念，记为 C=(A， B)，其中 B 和 A 分别被称为形式概念 C 的内涵(Intent)和外延(Extent)。设 C1=(A1，B1)和 C2=(A2，B2)是两个形式概念，其中偏序关系"≤"定义为 C1≤C2⇔B2≤B1。此时称 C1 是 C2 的子概念，C2 是 C1 的超概念。这样，由形式概念及父子关系就可形成一个概念格。

形式概念格是一个完全概念格，因此对于形式概念格中的任意结点子集，都存在唯一的最大下界和最小上界，分别使用

$$\bigwedge_{t \in T}(A_t, B_t) = \left(\bigcap_{t \in T} A_t, \left(\bigcup_{t \in T} B_t\right)''\right) \quad (3)$$

和

$$\bigvee_{t \in T}(A_t, B_t) = \left(\left(\bigcup_{t \in T} A_t\right)'', \bigcap_{t \in T} B_t\right) \quad (4)$$

来表示，这两个又成为形式概念的Meet和Join操作。

3 基于 FCA 的协作信息检索

将形式概念分析的理论知识应用到信息检索中，可以进一步提高检索的时间和空间效率。而对于多个数据库中进行协作式的检索，尤其是可以利用形式概念格中富含的信息进行 。基于文献[3]中提到的协作概念信息检索系统，它基于形式概念分析，提出了三个不同的检索系统，主要是对各个子形式背景进行约简，在约简对象的同时，获得等价对象集合，之后在对各个子形式背景检索得到一个临时结果集，再在此基础上，利用合并算法对这些临时结果进行合并，最终根据等价对象集获得最终检索结果返回给用户。但其中，有多点不足之处，比如，在子形式背景匹配时使用了精确的查找满足条件的对象集，但在实际情况下更需要的是一种模糊的，不精确的对象匹配过程。同样的，在对临时结果集进行合并时，合并组合各个临时结果，其匹配的过程也是精确的。另外一点不足之处就是，在使用形式概念格进行对象约简时，构造的形式概念格是一个完全格，这需要消耗非常大的时间空间资源，很多情况下构造完全格是没必要的。

3.1 协作信息检索结构框架

本文提出了一种改进之后的协作概念信息检索系统，针对多个子形式背景，分别构成子形式概念格，并同时生成等价对象集。对用户查询特定的关键词，在各个子形式概念格上，利用文献[7]提出的概念相似度计算公式，获得满足一定概念相似度阈值的临时形式概念集。然后将各个临时形式概念集根据合并算法进行合并，并同样适用前文提出的概念相似度对合并之后的形式概念进行衡量，符合概念相似度阈值的形式概念加入到最终结果集合。最终，在最终结果集合中每个形式概念的对象，包括与这些对象具有等价关系的对象即为最终检索结果返回给用户。整个过程可以由图 4 描述：

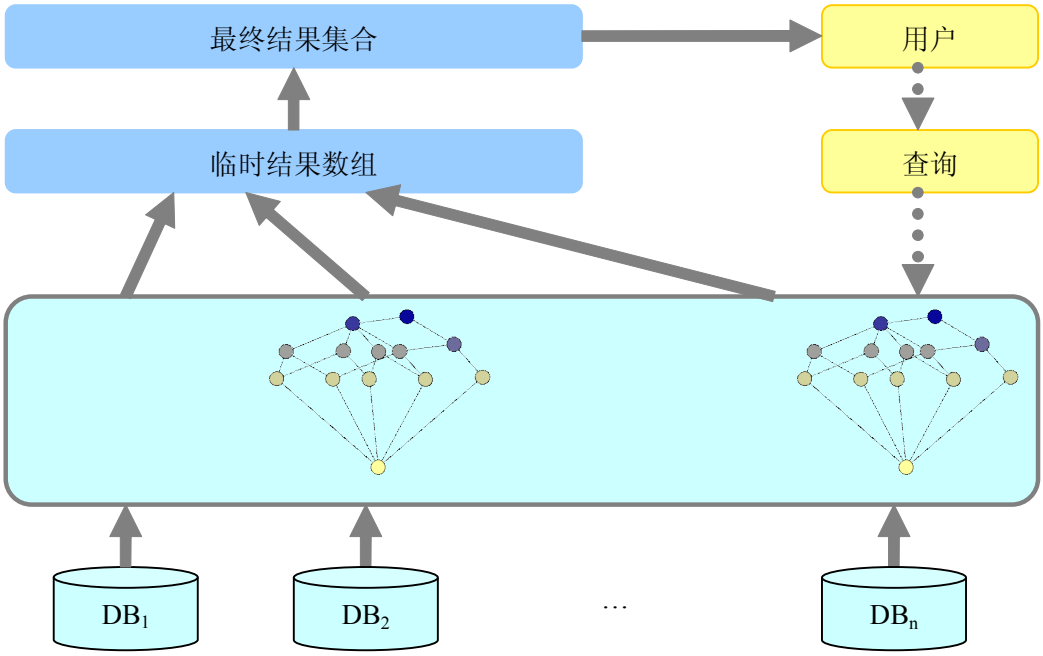

图 4 协作概念信息检索系统框图

从上图中，可以看到，对于各个不同的子数据库，构成各自独立的子形式背景，进一步的，根据相关构格算法构造子概念格，并在此过程中获得等价对象集合，相关的等价对象可以被约简，这形成了一个基本结构。当用户输入查询的各个关键词，分析得到对应的相关属性集，针对多个子形式概念格，在各自格上匹配满足一定相似度的形式概念集合并放入临时结果数组。这个过程中，由于各个子格上概念匹配的操作是独立的，所以非常便于分布式实现。当各个子过程完成后，再对存放在临时结果数组中的概念进行组合合并，并再一次进行相似度的匹配，最终获得的结果返回给用户。基于上述的整体思想，本文提出相关策略并设计分布式算法，实现了具体的细节。

3.2 等价对象集的约简

定义 5 closure[4,5]：假设 $x \in G$ 中的一个对象， $A \in G$ 是一个对象集合， I 为 $G \times M$ 上的一个关系，那么， $closure(x) = g(f(x)), closure(A) = g(f(A))$ 。

定义 6 等价对象集[3]：假设 $x \in G$ 中的一个对象， $A \in G$ 是一个对象集合， I 为 $G \times M$ 上的一个关系。对象 x 等价于对象集 A ，当且仅当， $\{x\} \bigcup A$ 是 I 上形式概念的外延， $closure(x) = closure(A) = \{x\} \bigcup A, x \notin A$ 。

比如，如下表 1 中的形式背景

	a	b	c
o1	×	×	×
o2	×		×
o3	×		
o4		×	×
o5			×

表 1 示例形式背景

其中，对象 o5 等价于对象集(o1，o2，o4)的，由于包含对象 o5 的形式概念是 CP((o1，o2，o4，o5)，(c))，而相反地，包含对象集{o1，o2，o4}的形式概念也是 CP。符合上述等

价对象集的定义，所以我们称对象 o5 是等价于对象集{o1，o2，o4}的。

既然对象 o5 可以由对象集(o1，o2，o4)等价表示，那么，可以在原来的形式背景中将对象 o5 约简，即对形式背景做对象层次上的约简，整个约简过程可由算法 1 描述：

算法 1：等价对象约简算法(FC)

输入：FC，原始形式背景

输出：经过对象约简后的形式背景

步骤：For 每个对象 o in FC：

p 为 o 对应的属性集合

查找形式背景中除对象 o 外的对象子集 sub，并得到其属性集 sub_p

If p == sub_p：

表示对象 o 可用对象子集 sub 来表示，即 sub 和 o 是等价的

将对象 o 从形式背景中删除，或者约简了对象 o 之后的形式背景 FC'

return FC'

算法 1 是针对概念对象进行的约简算法，而在属性层次上没有进行约简，这可在属性提取时进行，形成初步形式背景之后再根据此算法进行对象层次上的约简。这样从整体上就可以得到比较精简的形式背景。

3.3 协作信息检索过程

基于文献[3]中提出的协作信息检索系统的基本思想，本文提出改进了的协作信息检索系统，其过程可以分为三个步骤：

1) 针对各个子数据库 DB_i，经过相关预处理之后，形成初步形式背景 FC_i，并根据算法 1 对初步形式背景进行对象约简并得到等价对象集合。之后对约简后的形式背景使用渐进式构格算法 Godin 算法[6]生成子形式概念格 L_i。

2) 对于特定的查询词，抽取得到相关的查询属性集 T_1，T_2，…，T_n，其形式概念可记为((),(T_1，T_2，…，T_n))，即外延为空，在计算形式概念相似度时，可把它的层次号记为 0。然后分别在各个子形式概念格 L_i 上进行形式概念相似度的匹配，各个子临时结果集 SubConceptSet 保存入临时概念结果集 TempConceptSet。该详细过程可由如下算法 2 表示。

3) 在得到临时概念结果集之后，使用合并算法 3 将这些临时结果概念集进行组合，在进行相似度匹配之后，获得最终的概念集，其中各个形式概念的外延集合中的对象 FinalObjectSet，即为最终结果返回给用户。

其中，在各个子形式概念格上进行查找目标概念，得到子临时结果集 SubConceptSet，该算法是一个典型的可分布式执行的算法，时间和空间上都有一个很好的效果。

算法 2：子形式概念格概念匹配算法(L_i，(T_1，T_2，…，T_n)，sim_1)

输入：L_i 子形式概念格，(T_1，T_2，…，T_n) 查询属性集，sim1 为形式概念相似度阈值

输出：子形式概念结果集 $SubConceptSet_i$

步骤：子临时结果集 $SubConceptSet_i=\varnothing$

根据查询属性集(T_1，T_2，…，T_n)，找到一个子形式概念格 L_i 中，和形式概念((),(T_1，T_2，…，T_n)) 具有最大相似度的形式概念作为目标概念 C_i。

For 所有子形式概念格 L_i 中除 C_i 之外的形式概念 C_j 进行：

利用公式(15)计算 C_i 和 C_j 的概念相似度 Sim(C_i，C_j)

If Sim(C_i，C_j) > sim：

将形式概念 C_j)加入到子临时结果集 $SubConceptSet_i$ 中

return 子临时结果集 $SubConceptSet_i$

假如有 n 个子形式背景，分别构成 n 个子形式概念格，那么，利用算法 2 可以得到 n 个子临时结果集 $SubConceptSet_i$，该算法精简易懂更易于分布式的实现，所以在时间空间上都将会有较好的效率。

算法 3：概念合并算法($SubConceptSet_i$，(T_1，T_2，…，T_n)，sim_2)

输入：子临时结果集 $SubConceptSet_i$，i 为子形式概念格数，(T_1，T_2，…，T_n) 查询属性集，sim2 为形式概念相似度阈值

输出：最终概念对象集 FinalObjectSet

步骤：所有概念结果集 AllConceptSet=∅

For 每个子临时结果集 $SubConceptSet_i$ 进行：

将各个子临时结果集并入所有概念结果集

AllConceptSet= AllConceptSet⋃ $SubConceptSet_i$

For 所有概念结果集 AllConceptSet 中的每个概念 C_i 进行：

进行两两组合得到新的形式概念

For 所有概念结果集 AllConceptSet 中除 C_i 外的其他形式概念 C_j 进行：

新的概念的内涵为 C_i 内涵和 C_j 内涵的并集，外延为 C_i 外延和 C_j 外延的交集

NewConcept(i，j)=$((\text{extent}(Ci) \cap \text{extent}(Cj), \text{intent}(Ci) \cup \text{intent}(Cj)))$

计算得到新的形式概念 NewConcept 和形式概念 T((，(T_1，T_2，…，T_n)) 的概念相似度 Sim(NewConcept，T)。

If Sim(NewConcept，T)大于 sim2：

将 NewConcept 的外延加入到最终概念对象集 FinalObjectSet 中

return 最终概念对象集 FinalObjectSet

通过算法 3，我们能够获得满足一定相似度阈值的最终概念集合，这些概念集合中每个形式概念的外延即为满足用户查询条件的对象集。其中，算法 2 和算法 3 中都涉及到两个相似度阈值，分别为 sim_1 和 sim_2，在对子形式概念格进行相似度的匹配时，由于子形式概念格中具有的内涵总数可能不多，可能也不完全，所以对于 sim_1 应该设定的小一点以便不会遗漏可用的形式概念，而在合并时，形式概念的合并，由于是通过其内涵的合并进行的，相对来说其内涵信息丰富，所以设置 sim_2 应该稍大一点以便过滤掉不匹配的形式概念。

3.4 实例描述

为了更好的说明本文提出的协作信息检索系统的有效性，以下进行简单实例描述。

假设，现有三个形式背景，分别如表 2 所示。

	a	b	c
o1		×	
o2	×		×
o3		×	×
o4			×

	a	b	d
o1		×	×
o2	×		×
o5			×

o6	×		
o7		×	×

	a	c	d
o2	×	×	×
o5			×
o6	×		

表 2 三个子形式背景

三个子形式背景经过算法 1 可以得到约简后的子形式背景如下表 3，并可以得到等价对象集合为：o4={o2，o3}，o1=o7。

	a	b	c
o1		×	
o2	×		×
o3		×	×

	a	b	d
o1		×	×
o2	×		×
o5			×
o6	×		

	a	c	d
o2	×	×	×
o5			×
o6	×		

表 3 进行对象约简后的三个子形式背景

接着对这些子形式背景使用 Godin 算法，构成子形式概念格，如图 5 所示：

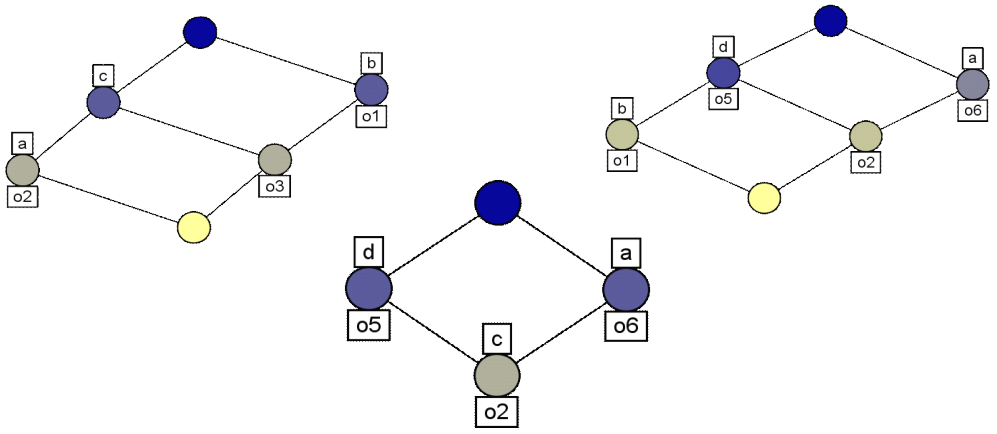

图 5 生成的各个子形式概念格

假如，用户需要查询属性集为(a，d)，针对各个子形式概念格，按照算法 2 的思想进行查找，其中设定形式概念相似度 sim_1 为 0.5。这里由于属性的语义无法衡量，所以在利用公式(15)时，设定 ω 为 1，即只考虑形式概念的结构相似度，但在实际过程中考虑形式概念的语义相似度是非常有必要的。

在第一个子形式概念格中，找到和形式概念((),(a，d))最相似的形式概念为 C((o2)，(a，c))， 它们之间的相似度为 0.5，并把这个形式概念 C 作为该子形式概念格中的目标形式概念，接着使用这个目标形式概念和格中其他形式概念进行相似度的计算，满足 sim_1 的形式概念加入到子临时结果集 $SubConceptSet_1$，可以找到，形式概念((o2，o3)，(c))和形式概念 C 的相似度为 0.75，形式概念((o3)，(b，c))和形式概念 C 的相似度为 0. 5，它们都满足相似度阈值，所以并入子临时结果集 $SubConceptSet_1$={((o2)，(a，c))， ((o2，o3)，(c))， ((o3)，(b，c))}。按照同样的思想，分别对第二，三个子形式背景进行计算，得到 $SubConceptSet_2$={((o2)，(a，

d))，((o1)，(b，d))，((o1，o2，o5)，(d))，((o2，o6)，(a))}和子临时结果集 $SubConceptSet_3$={((o2)，(a，c，d))，((o2，o6)，(a))，((o2，o5)，(d))}。这样得到所有可能的概念集合为{((o2)，(a，c))，((o2，o3)，(c))，((o3)，(b，c))，((o2)，(a，d))，((o1)，(b，d))，((o1，o2，o5)，(d))，((o2，o6)，(a))，((o2)，(a，c，d))，((o2，o6)，(a))，((o2，o5)，(d))}。经过算法 3 进行合并成新的形式概念，总共有{((o2)，(a，c，d))，((o3)，(b，c)，((o1)，(b，d))}。假定 sim_2 为 0.6，计算这些概念和形式概念((),(a，d))相似度，可以分别得到相似度为 0.666，0.5，0.5，那么，满足相似度阈值的形式概念为((o2)，(a，c，d))，它的外延就是 o2，再结合等价对象集，形成最终对象集返回给用户，即在这个例子中仍然是 o2。

4 复杂度分析

假设现有 n 个子形式背景，通过 Godin 算法得到子形式概念格，其时间复杂度为 $O(2^{2K}|G|)$，G 为对象的个数，K 一般情况下为一个固定值。等价对象约简算法，即算法 1 需要的时间复杂度为 $O(|C|2^M)$，C 为形式概念格中的所有形式概念，M 为属性个数。子概念格上形式概念的匹配算法，即算法 2，仅需要遍历整个形式概念集即可找到满足一定相似度阈值的形式概念，即时间复杂度为 $O(|C|)$。对于合并算法 3，由于需要对形式概念进行两两合并，如果临时概念结果集中有 N 个形式概念，那么合并的时间复杂度为 $O(\frac{N^2}{2})$。综合考虑，整个过程需要的时间复杂度为 $O(2^{2K}|G|+|C|2^M+\frac{N^2}{2})$。

但从整体上考虑的话，本文提出的整个系统能够很方便的部署在分布式计算环境中，那么，整个系统性能会在时间和空间上都将有一个较大的提升。

5 总结与展望

在协作信息检索中，采用对各个子形式背景分别利用形式概念相似度的衡量方法对已有形式概念进行匹配，在找到临时概念集之后采用合并的方式，获得新的概念，再对新的概念进行相似度的匹配，最终获得满足用户需求的结果集合。整个系统充分体现了协作的思想，便于在分布式环境中部署实施，并且在匹配查询关键词时使用了不精确的方式，从结构和语义两个层次上进行衡量，更好的体现了人性化需求。但本文虽然提出了整个系统框架和具体的算法，但由于硬件条件的限制，使得整个系统无法在真正分布式的环境下实现，所以在最后的评测分析中无法得到现实数据来分析性能，而仅仅在理论上分析时间空间复杂度。这会在以后的工作中继续完善，得到更加详细的数据结果报告以便做进一步的改进。

参考文献:

[1] Uta Priss. Formal Concept Analysis in Information Science. Knowledge-Based Systems archive Vol. 21 ， Issue 1 (February 2008) Pages 80-87

[2] B.Ganter ， R.Wille. Formal Concept Analysis: Mathematical Foundations. Springer-Verlag， 1999.

[3] Nafkha I.， Elloumi S.， Jaoua A.， Using Concept Formal Analysis for Cooperative Information Retrieval. Concept Lattices and their applications Workshop (CLA'04)， VSB-TU Ostrava， September 23th-24th， 2004.

[4] Jaoua A.， Bsaies Kh.， and Consmtini W. : May reasoning be reduced to an Information Retrieval problem. Relational Methods in Computer Science， Quebec， Canada， (1999).

[5] Jaoua A.， Al-Rashdi A.， AL-Muraikhi H.， Al-Subaiey M.， Al-Ghanim N.， and Al-MisaifriS.: Conceptual Data Reduction， Application for Reasoning and Learning. The 4th Workshop on Information and Computer

Science， KFUPM， Dhahran， Saudi Arabia， (2002).

[6] Godin R,MiLi H,Mineau GW,Missaoui R, Arfi A , ChauT-T. Design of class hierarchies based on concept (Galois) lattices[J]. Theory and Application of Object Systems, 1998, 4(2):117-134

[7] Sheng Yan, Li Yun, Luan Luan. A Concept Similarity Method in Structural and Semantic Levels. Second International Symposium on Information Science and Engineering, 26 - 28,Dec. 2009 Shanghai, China 这个页面还不知道