本科毕业论文



|  |  |  |  |
| --- | --- | --- | --- |
| 论文题目： | 基于知识图谱的关系概念化 | | |
|  |  | | |
| 院系： | 软件学院 | | |
| 专业： | 软件工程 | | |
| 姓名： | 范思奇 | 学号： | 11302010045 |
| 指导教师： | 肖仰华 | 职称： | 副教授 |
| 单位： | 复旦大学 | | |
| 日期： | 2015 年 06 月 11 日 | | |

目录

摘要 1

Abstract 2

1. 引言 3

1.1. 背景 3

1.1.1. 知识图谱 3

1.1.2. 实体关系 4

1.1.3. 实体关系抽取 5

1.2. 研究内容与主要贡献 6

2. 相关工作 8

2.1. 早期关系抽取中的关系分类体系 8

2.1.1. 早期关系分类方法与封闭领域关系抽取 8

2.1.2. 早期关系分类与抽取的局限性 8

2.2. 基于知识库的关系抽取中的关系分类体系 8

2.2.1. 基于知识库的关系分类和关系抽取系统 8

2.2.2. 现有实体关系分类体系的局限性 11

3. 研究内容 12

3.1. 定义 12

3.1.1. 概念分类体系 12

3.1.2. 实体关系 12

3.1.3. 常用符号表 13

3.2. 问题描述 13

3.2.1. 研究目标 13

3.2.2. 问题的难点 15

3.3. 算法设计 16

3.3.1. 概念对生成 17

3.3.2. 聚类压缩 21

4. 实验 26

4.1. 数据处理 26

4.2. 准确率 26

4.3. 聚类效果 28

4.4. 质量评估 31

5. 结论 33

5.1. 研究结论 33

5.2. 应用 33

5.2.1. 基于语义的关系分类体系 33

5.2.2. 基于语义关系的模板匹配 34

5.3. 改进方向 34

5.3.1. 概率方法 34

5.3.2. 多元关系 35

5.3.3. 偏移问题 35

6. 参考文献 36

7. 致谢 38

摘要

随着一些质量高、体积大的知识图谱的出现，信息抽取工作获得了更多的语义知识。而基于知识图谱的实体关系抽取仍然处于很直观的、初级的阶段，其所面临的关键问题就是关系识别与关系分类问题。目前的主要工作是基于实体本体的，针对一对固定好的、粗粒度的实体类型，对实体关系进行句法分析的关系分类，或是聚类发现新关系。为了从更细粒度的实体类型上描述实体关系，本文提出了基于概念分类系统的，两步抽象自下而上的关系概念化方法。对给定的一个实体关系，根据概念的典型性、多样性、覆盖率等特征为其给出一组Top-K的概念对，这组概念对可以用于抽象描述这个实体关系。实验表明，本文提出的算法可以发现一个实体关系下粒度更细的关系，也可以从一个组合实体关系中以较高的准确率分离关系，同时本算法生成的用于描述关系的概念对具有较高的质量。

**关键词：知识图谱 实体关系 概念化 聚类分析**

Abstract

As numbers of high-quality, large volume knowledge graphs appear, information extraction work has been enriched with more semantic knowledge. However, the entity relation extraction based on the knowledge graph is still at a very intuitive early stage, and the key issue it faces is the relation recognition and classification. Current works are mostly based on manually constructed entity ontology, where coarse-grained entity types are fixed for syntactic analysis on classification or clustering on detecting new relations. In order to break the shackles of ontology and describe the relationship between the entities with fine-grained types, we propose a two-step bottom-up abstraction approach for relation conceptualization based on conceptual taxonomy that is automatically constructed. Given an entity relation, we figure out a group of Top-K concept pairs to abstract the relation, according to the typicality, diversity and coverage features. Our experimental evaluation shows that our method performing significantly high precision and quality for detecting fine-grained relationships.

**Key words：Knowledge graph, Relation conceptualization, Clustering**

2. 引言
   1. 背景

互联网中存在大量的非结构化或半结构化文本，包括新闻、博客、政府公文、聊天日志等。为了让计算机理解这些数据以便提供更准确、智能的服务，一个主流观点是将这些非机构化半结构化文本转化成结构化的语义信息，这个工作就是信息抽取（IE，Information Extraction），包括实体抽取（Entity Extraction），实体关系抽取[[1]](#footnote-1)（Relation Extraction）和事件抽取（Event Extraction）。

在信息抽取工作之初，由于信息结构需要专家人为定义，在有限成本的情况下，针对特定领域的有限信息抽取成为当时的主流方法。而如今，随着一些质量高、体积大的知识图谱（Knowledge Graph）的出现，如DBpedia，Freebase，YAGO，Probase等，信息抽取工作有了新的研究方向。这些知识图谱包含了自动构建的海量实体（Entity）、概念（Concept）、语义关系（Semantic Relation）等数据。作为背景知识，这些数据可以被充分利用到信息抽取的过程中：以往抽取出的结构化信息将被关联到知识图谱中，而知识图谱自身的关联关系将大量的抽取信息构建成更庞大的结构，从而促进新的信息抽取，如此迭代优化下去，使信息抽取工作变得更加自动化且覆盖更广泛的领域。

通过分析根据以上背景，基于知识图谱的信息抽取需要解决的关键问题是：1）如何运用知识图谱抽取信息，2）如何将抽取出的信息关联到知识图谱中。

当前的实体抽取已经针对这两个问题提出了许多关于解决方案，而基于知识图谱的实体关系抽取仍然处于很直观的、初级的阶段。

* + 1. 知识图谱

狭义地说，知识图谱概念是指由谷歌公司提出的支持智能化搜索引擎的知识图谱。而学术界近年来普遍接受一个广义的知识图谱定义，即“知识图谱本质上是一种语义网络，其结点代表真实世界中存在的各种实体或概念，边代表实体/概念之间的各种语义关系。” [[2]](#footnote-2)

知识图谱与如本体论（Ontology）、语义网（Semantic Web）的区别在于：本体论通常由树结构表示，其不支持一个子结点拥有多个父结点，而现实情况中一个词语在不同场景下可以具有不同的概念，如“苹果”可以代表苹果公司，也可以代表水果，然而本体论如果需要解决这个问题就会产生很多冗余信息，而对于知识图谱来说，有向无环图的分类体系构建就可以轻松胜任；而语义网通常用于描述各类事实，作为一种概念性，语义网更多侧重于知识的表达，而不是知识的存储和管理。

早在知识图谱出现之前，知识库（Knowledge base）这一概念已经被广泛使用，如DBpedia，Freebase等。区别于原始的由专家系统中的知识库，这些知识库的构建是对应现实世界中存在的开放领域（Open Domain）信息。知识图谱作为一种进阶版的知识库，其特点是覆盖率和自动化程度更高，同时其语义关系也更加复杂、全面、准确。随着知识库的不断完善和自动化程度的不断提高，越来越多的知识库被称作为知识图谱。因此，本文中提及的部分知识库也均指广义上的知识图谱，只不过由于知识图谱概念提出较晚，故根据时间顺序部分内容中依然称作知识库。

* + 1. 实体关系

实体关系（Entity Relation）是指某一时间段内实体之间存在的关系。一个关系r被定义为一个元组的形式t=(e1, e2,…, en)，其中ei是文档中与r有关联的实体。二元关系的例子包括located-in(Fudan University, Shanghai)，ceo-of(Tim Cook, Apple Company)。此外还有一些多元关系，如句子“At codons 12, the occurence of point mutations from G to T were observed”[[3]](#footnote-3)中包含了一组四元生物关系，即point\_mutation(codon, 12, G, T)。

根据美国国家标准技术研究院（National Institute of Standards and Technology，NIST）的自动内容抽取（Automatic Content Extraction，ACE）评测，局部（Local Relation Detection & Recognition，LRDR）关系检测和识别与全局关系检测和识别（Global Relation Detection & Recognition，GRDR）共包括7种实体关系，其中每一个大类又包含多个子类型：

|  |  |
| --- | --- |
| 实体关系类型 | 子类型 |
| 制造使用关系（Agent-Artifact） | 用户-所有者-发明人-制造者  User-Owner-Inventor-Manufacturer |
| 类属关系（General-Affiliation） | 公民-居民-宗教-种族  Citizen-Resident-Religion-Ethnicity  组织-位置  Org-Location |
| 转喻关系（Metonymy） | 无 |
| 组织结构从属关系（Organization-Affiliation） | 雇佣，创立者，所有者关系，研讨生，  运动-附属关系，投资者-股东，成员关系  Employment, Founder, Ownership,  Student-Alum, Sports-Affiliation,  Investor-Shareholder, Membership |
| 局部整体关系（Part-Whole） | 人造品局部整体关系，地理局部整体关系，  附属关系  Artifact, Geographical, Subsidiary |
| 人物关系（Person-Social） | 商业，家族，长期个人关系  Business, Family, Lasting-Personal |
| 地理位置关系（Physical） | 位于关系，邻近关系  Located, Near |

表 1 NIST自动抽取关系分类

然而，这7类实体关系非常粗糙，只能用于泛泛地理解实体关系的含义，而且7种关系之间仍有相互重叠的部分，也没有覆盖所有已出现的实体关系，因此在实际的实体关系抽取中没有得到应用。

* + 1. 实体关系抽取

实体关系抽取，是指从文本中抽取包括两种数据的抽取：

* **关系模式（Relation/Relational Pattern）**

具有某种关系的一组实体，通常会与某些常出现的上下文（context）组合后出现在文本中，这些常出现的上下文就是关系模式。如“Arg1位于Arg2”，“Arg1担任了Arg2的Arg3”等。关系模式中的占位符Arg#表示的是一个实体，而通常在关系模式中，实体的类别需要限定，因此一个完整的关系模式应该如“<Person> write a song <Song>”。一些更高级的关系模式包含了等价语句，因而可以代表一类关系模式，如“<Person> write a [adj] song <Song>”

* **关系实例（Relation Instance）**

关系实例是指具有某种关系的一组实体，这组实体与关系模式相互对应。

可以发现，关系模式和关系实例是相辅相成的，关系模式可以用于抽取关系实例，而关系实例也可以用于从文本中发现关系模式。通常我们所说的实体关系抽取，就包括了关系模式和关系实例两个部分的抽取。实体关系的抽取依赖于实体的识别，因此只有当实体识别具有一定的准确率之后，实体关系抽取工作才能得意进展。

无论是关系模式还是关系实例，抽取出来之后都要有一个良好的组织结构用于储存，以便检索和使用。关系实例作为三元组存储在知识图谱中，而关系模式存储在关系分类体系中。关系分类体系可以是人工建立的简单扁平关系分类，或者稍复杂的多层树形关系分类，也可以是基于关系模式句法、语义特征而自动构建的关系分类体系，还可以是基于关系模式中实体类型的关系分类体系。

* 1. 研究内容与主要贡献

如今实体识别、实体抽取、实体关联工作已经具有较高的准确率、覆盖率和数量级，一直处于直观、初级阶段的实体关系抽取成为信息抽取的重要研究领域。实体关系抽取的主要作用是1）为机器智能提供语义关系的背景知识，2）促进实体类型的识别与抽取。当前关系抽取所面临的关键问题就是关系识别与关系分类，而这两个问题都依赖于关系分类体系。早期关系抽取工作（如KnowItAll[4]，S Schiff 2006）的关系分类体系都是人工构建的扁平或两层关系关系分类体系；NELL（Carlson 2010[14]；Mohamed 2011[15]）的关系抽取是与实体抽取相互迭代促进的，其关系分类体系是根据关系模式而人工构建的；PATTY（MPI 2012[20]）是根据关系模式的句法和语义关系自动构建的一个关系分类体系。

然而目前的上述工作中依然存在如下问题：

问题1：关系的识别和分类主要基于人工构建的本体，其所包含的实体类型有限，且只能在一对固定好的、粒度较粗的实体类型下分析实体关系模式而进行关系识别或分类，进而抽取的实体关系只覆盖少量粒度较粗的实体类型。

问题2：不同实体类型组合的新关系难以发现。以往工作中新关系的发现是通过关系模式聚类来探测同一对类型的实体之间的新关系，因而新关系的发现也局限于原本人工构建的关系分类体系中的实体类型。

目前大量概念分类体系（Conceptual Taxonomy）的出现为解决这一问题带来了契机。概念分类体系因为其自动构建的特性，包含丰富实体的类型，使得关系抽取中特定关系模式中的实体类型可以变得更具体，粒度更细，因而能够极大地丰富关系分类体系的内容，同时促进更细化的实体类型识别。

因此，本文研究的目标正是如何利用知识图谱中的概念分类体系，为一个实体关系找出最恰当的一组概念对，用于识别实体关系，并发现实体类型粒度更细的新实体关系。本文提出了两步抽象自下而上的关系概念化方法（图1），实验结果表明，本文提出的算法可以发现一个实体关系下粒度更细的关系，也可以从一个组合后的实体关系中以较高的准确率分离关系，同时本算法生成的用于描述关系的概念对具有较高的质量。

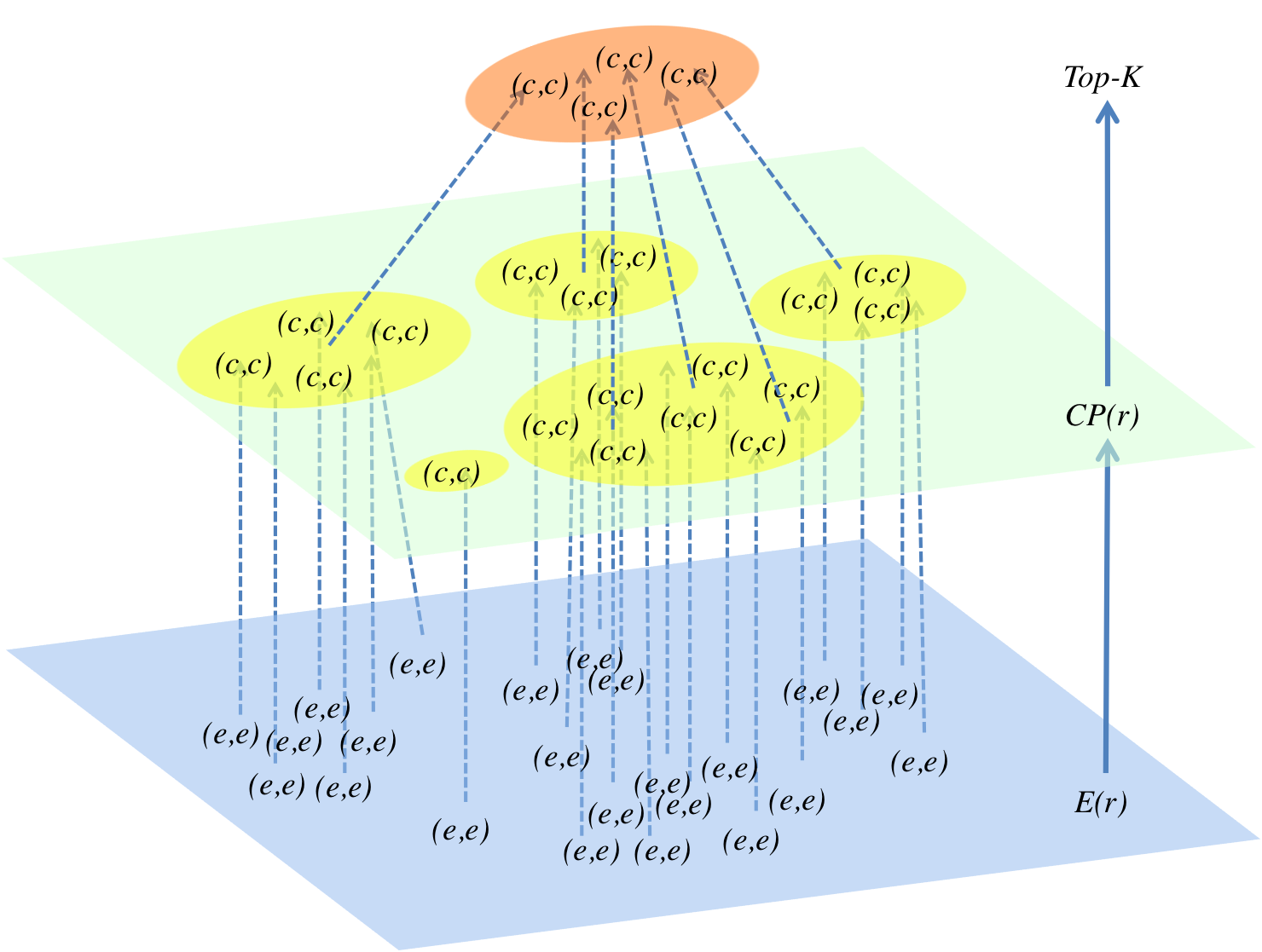


图 1 两步抽象自下而上的关系概念化方法

1. 相关工作
   1. 早期关系抽取中的关系分类体系
      1. 早期关系分类方法与封闭领域关系抽取

早期的关系抽取工作都是基于人工定义的关系分类来抽取。Rosario和 Hearst（2001）将药品领域下的关系分为13类[5]；Stephens等人（2001）将基因关系抽取到17个特定类中[6]；Nastase和Szpakowicz （2003）提出了双层关系结构，其中第一层包括5类，第二层包括30类，用于抽取名词修饰语关系[7]；还有大量工作（Kim and Baldwin, 2005[8]; Nakov and Hearst, 2008[9]; Nastase et al., 2006[10]; Turney and Littman, 2005[11]）都是针对某个特定领域或常识的关系分类。

基于早期的关系分类方法，关系抽取主要运用监督学习算法：将关系抽取当做一个分类问题来处理。主要包括基于特征的方法（Feature based methods）或核方法（Kernel based methods）。学界统一将这类关系抽取称为传统关系抽取[3]，或称封闭领域关系抽取（Closed Relation Extraction）。

* + 1. 早期关系分类与抽取的局限性

传统关系抽取是早期提出的关系抽取模型。由于其关系分类是专家挑选的特征来进行训练的，针对特定领域的小范围抽取是比较高效的。

然而，传统关系抽取仍然有一些局限性：

* 难以扩展新的实体关系类型
* 难以抽取多元关系
* 关系抽取的特征难以权衡和优化
* 由于依赖人工标记数据，传统关系抽取不能处理大规模的数据

而抽取系统最主要的一个问题就是，即便是针对同一领域，不同系统抽取出的实体关系难以统一，而不同领域的关系分类由于衡量方式不同，也会相互冲突。

因此，这类基于早期实体关系分类而抽取出的实体关系不具备普遍意义，甚至只是纯粹的文本而已。

* 1. 基于知识库的关系抽取中的关系分类体系
     1. 基于知识库的关系分类和关系抽取系统

知识库拥有丰富的命名实体分类体系（Taxonomy），然而却只有少量的关系和简单的关系分类方法：Freebase最初只有几千个二元关系实例；DBpedia也只有8000多个从Wikipedia中获取的属性关系类型，而其中还包括大量的冗余，同时对一个关系类型来说也只涉及关系实例，而不涉及关系模式。因而在2007年之后，为了自动扩充知识库，基于知识库的关系抽取系统陆续出现。

大部分基于知识库的关系抽取系统都使用无监督或半监督方法，有封闭领域的也有开放领域的。著名的基于知识库的关系抽取包括：TextRunner/ReVerb（Banko 2007[12]; Fader 2011[13]），NELL（Carlson 2010 [14]；Mohamed11[15]），Probase（Wu 2011[17]），动态词汇表抽取方法（Hoffmann 2010[18]），LDA聚类方法（Yao 2011[19]），PATTY（Nakashole 2012[20]；Nakashole 2013[21]）。

根据实体关系分类体系的构建方式不同，本文将介绍NELL和PATTY两个系统的相关工作。其中NELL的关系分类体系是是基于实体本体论（entity ontology）而人工构建的，其后续升级的OntExt部件则提供了自动关系发现的功能；PATTY是基于关系模式的句法、语义特征而构建的一个层次丰富的关系分类体系，其自动构建的特点使得这个关系分类体系更庞大、更全面，同时由于PATTY是根据关系模式构建的，其能够适用于更广泛的关系抽取平台以抽取海量关系实例。

* + - 1. NELL/OntExt

基于知识库的信息抽取系统通常都是半监督方法或自举方法（Bootstrapping），前提是需要有可靠的种子实例，包括实体实例和关系实例，但部分少量的人工标记数据中经常存在一些不可信的数据，这些种子会影响之后一些列的迭代抽取过程。为了克服这个问题，卡内基梅隆大学机器学习系的科学家提出了耦合式半监督学习方法（Coupling Semi-Supervised Method）[16]，并于2010[14]年推出了NELL（Never Ending Language Learning）系统[[4]](#footnote-4)。

耦合式半监督学习方法的主要思想是：基于本体论的约束，使用不同类别、不同关系的多个分类器进行同步学习。由于同一个本体论中不同的类别与关系之间有相互的联系，不同的分类器之间就会产生相互约束的作用，只要保证大部分种子是可靠的，那些不可靠的数据就会在迭代中被这种全局约束排除掉：单一的分类器会由于人工标记数据的不可靠从而失灵，而多个分类器的共同作用则能使不可靠的数据逐步被弱化，从而提高整体的准确度。因此，NELL系统是同时抽取命名实体和实体关系的系统。

NELL抽取关系所依赖的关系分类体系是根据实体本体论而人为构建的，比如personHasCitizenship:<person>×<country>是一个二元关系，其祖先路径是personHasCitizenship->agentRelatedToLocation->atLocation->relatedTo。人为构建关系分类体系中关系已经预知，所以不存在关系关联（relation linking）的问题，而其缺陷也在于关系无法扩充，难以学习新的关系。

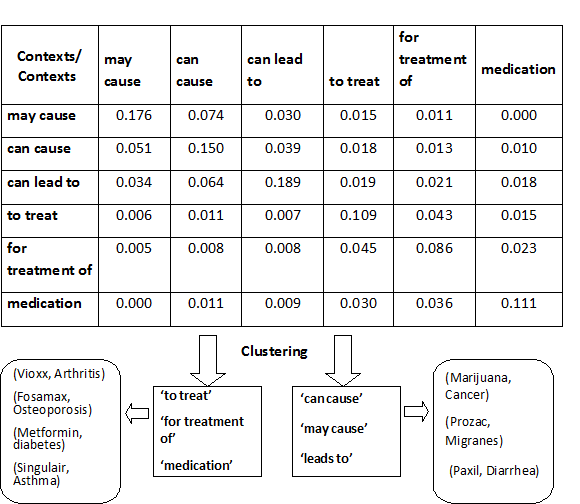


图 2 OntExt聚类示意图

为了克服这个问题，NELL后续又增加了OntExt[15]部件，通过上下文聚类的方式，对给定一组实体类型进行聚类，从而产生不同的关系模式簇而获得新的关系类型（图2所示）。然而对于一组实体类型来说，其产生的上下文数量非常庞大，所以OntExt仍然难以应对大规模语料的关系抽取。本文提出的方法与OntExt最为相似，OnExt是在同一对实体类型下对关系模式的聚类，本文是在不同实体类型间对关系实例的聚类而产生新关系。

* + - 1. PATTY

PATTY是基于关系模式的句法、语义特征，自动构建的类似于WordNet[24]的二元关系分类体系，由德国马克斯-普朗克研究所（MPI，Max Planck Institute）于2012年提出[20]，紧接着MPI于2013年推出了基于PATTY的关系抽取系统[21]。

MPI于2012年的论文中为PATTY的构建提出了句法-本体-词汇模式（SOL，syntactic-ontologic-lexical patterns），该模式包含了词序、POS标签、通配符和本体类型。一个SOL模式就像一个关系模式一样，可以用来匹配抽取关系实例。如“person’s [adj] voice \* song”可以从“Elvis Presley’s solid voice in his song ‘All shook up”中抽取到(Elvis, AllShookUp)这个关系实例。SOL模式可以从种子关系实例抽取到的上下文中根据依存分析建立，因此符合某种句法形式的SOL模式就可以根据规则分为一类，而更复杂的句法形式则属于子类。此外一些关系模式实际上并不存在句法关系，需要用语义的隶属度来表示父子类关系，因此PATTY系统中定义了弱包含关系，用来描述语义上两个关系模式的隶属度。然而当S集合比较小时，S的这种隶属度的置信度不够高，可能仅仅是因为S比较小而其元素恰好大部分落在B中。所以PATTY引入了威尔逊分数（Wilson score）的概念，从概率的角度考虑了样本大小的置信度问题：假设S是真实的S’集合的一个样本，S’中的元素属于B的概率是可以靠S中元素属于B的概率来估计的。样本的概率落入[c-d, c+d][[5]](#footnote-5)中，如果S较小则c趋近与0.5，d比较大，S较大则c趋近于，d比较小。通过比较置信区间的下限就能估计小样本的隶属度了。



通过分析句法和语义的隶属关系，PATTY将语料中抽取出的SOL模式进行关系分类体系构建：根据关系模式的频率排序构建前序树（FP Tree），然后将SOL模式关联到前序树后用类似于平凡模式挖掘的方式构建有向无环图，成为每个节点都是SOL模式的关系分类体系。

* + 1. 现有实体关系分类体系的局限性

现有的实体关系分类体系主要存在以下两类问题：

问题1：关系的识别和分类主要基于同一对实体类型下的关系模式，同时因为本体是人工构建的，其所包含的实体类型有限，从而抽取出的关系只覆盖少量粒度较粗的实体类型。比如PATTY系统的关系模式使用的实体类型是DBpedia和YAGO两个实体本体，因而其构建仍然受限，而且关系分类体系构建是一次性完成的，因而无法自增。

问题2：不同实体类型组合的新关系难以发现。以往工作中新关系的发现是通过关系模式聚类来探测同一对类型的实体之间的新关系，因而新关系的发现也局限于原本人工构建的关系分类体系中固定好的的实体类型。比如OntExt中发现的新关系对于促进实体识别来说没有作用，因为通过新的关系模式发现的新关系的实体类型仍然没有变化。

1. 研究内容
   1. 定义
      1. 概念分类体系

***概念-实体*：在概念分类体系中中，概念是实体的抽象表示，实体是概念的具象实例**。

一种实体和*概念*之间的关系使用*isA*边来表示的。如isA(apple, company)、isA(apple, fruit)。知识图谱中由*isA*边构成的子图是一个有向无环图。现有的最大的概念分类体系是微软在2012年发布的Probase[26]，其特点是涵盖的概念数量是目前所有知识图谱中最多的（约2.6M），同时其isA关系是基于概率的。

* + 1. 实体关系

***实体关系*：实体关系实际上是对许多*实体对*（entity pair）的抽象表示**。比如writer这个关系，在DBpedia中包括许多如*实体对*，如(The Angelic Conversation*[[6]](#footnote-6)*, William Shakespeare)，(Heal the World, Michael Jackson)等。这种抽象程度越高，关系能覆盖的*实体对*就越多；反之，抽象程度越低，关系就越具体，覆盖的*实体对*就越少。因此，关系的分类体系，实际上是一种抽象层次，而构建的过程实际上一个逐步抽象的过程。

我们认为，**一组由许多*实体对*抽象化得到的*概念对*（concept pairs）可以用于表示某种关系**。比如writer关系可以表示为一个*概念对集合*[(song, artist), (firm, director), (book, writer)]。进一步地，我们发现一个*概念对集合*中的子集可以表示一种子关系，如[(book, writer)]可以表示author这样的子关系，而如[(song, artist), (firm, director)]则可以表示multimediaWriter这种新的关系。传统的实体关系的描述实际上是定义一个词，对于计算机来说，很难理解这个词背后的语义是什么；而如果直接用一组实体对去表示，往往数量又太多。因此我们提出，将一个关系下的*实体对集合*进行概念化，即*关系概念化*（relation conceptualization），或*实体对概念化*（entity pair conceptualization），是有助于关系分类、新关系发现以及关系补全。

这里值得讨论的是，除了概念分类体系中的*概念*，还有如*分类*（category）、*本体类别*（owl:type）都可以抽象地描述一个实体。我们之所以不使用*分类*的原因是，*分类*不具备isA关系，其描述的是一种相关性。比如William Shakespeare在Wikipedia中的*分类*有Shakespeare family，但二者之间并不是isA关系，而是一种从属关系，而从属关系的两个实体各自具备的实体关系并不能传递，比如writer(The Angelic Conversation, William Shakespeare)不能抽象化为writer(The Angelic Conversation, Shakespeare family)。我们不使用*本体类别*的原因是，本体论中一个实例（instance）的本体类别只有一个本体类，然而对于某些实体如Patti Smith这个人同时写过书和歌曲，其抽象描述既可以是作者又可以是歌手，而本体论中对于这类实体的表示要么存在冗余，要么存在歧义，进而在抽象描述关系，特别是新关系的时候，消岐工作就会变得很复杂。

与*分类*和*本体类别*不同，概念分类体系中的*概念*对*实体*进行抽象化的方式是isA关系。实体和*概念*间的isA关系能够确保当实体被抽象之后，实体关系仍然保留；同时，基于概率的概念分类体系中每条isA边对应一个概率，一个实体可以以不同的概率抽象为多个不同的*概念*，这些概率非常有助于*实体对抽象化*的过程中的筛选和消岐。因此，我们使用知识图谱中的*概念*作为实体抽象化表示，继而提出前文中说到的*关系概念化*。

* + 1. 常用符号表

这里我们定义一些知识图谱中常用的符号。

|  |  |
| --- | --- |
| 符号 | 解释 |
| G | 知识图谱，结点为实体，边为实体关系 |
| e | 命名实体，本文中简称实体 |
| E | 知识图谱中的边集合 |
| r | 命名实体关系，简称实体关系 |

表 2 符号定义表

* 1. 问题描述
     1. 研究目标

给定一个知识图谱G，其中结点为实体或概念e，边为x(ei, ej)，表示实体ei和ej间存在一条关系r的边。对一个二元关系r，在知识图谱G中存在一个*实体对集合*。将E(r)中的*实体对*概念化之后的集合称为关系的概念对集合CP(r)。



例 1. *表3中有关系writer的一些实体对的一些例子，这些实体对概念化之后被分成了三个不同的概念对。*

|  |  |  |
| --- | --- | --- |
| r | E(r) | CP(r) |
| writer | (The Beautiful People, Marilyn Manson) | (song, artist) |
| (Heal the World, Michael Jackson) |
| (Fly High, Ayumi Hamasaki) |
| (Yamato, Junya Sato) | (film, director) |
| (Metropolitan, Whit Stillman) |
| (Sky High, Tsutomu Takahashi) |
| (The Angelic Conversation, William Shakespeare) | (book, writer) |
| (The\_Secret\_Book, Jordan\_Plevnes) |

表格 3 关系writer的例子

显然，我们希望CP(r)尽可能小就足以描述一个关系r。CP(r)中的每一个*概念对*需要有足够的典型性能够抽象E(r)中的某个子集，同时所有的*概念对*总体的覆盖率要足够高，即多样性要足够丰富。

这里我们需要说明典型性（typicality）和覆盖率（coverage）的含义。典型性是指当我们用一个概念去抽象描述一个实体时，这个概念的语义不能太宽泛，也不能太罕见。比如一个实体Michael Jackson，一个典型的概念是artist，一个宽泛的概念是person，一个罕见的概念是cultural leader。在知识图谱中，这种典型性通过一个频数来表达，即某个实体以某个概念呈现在一定预料中的次数。覆盖率是指一个概念能够抽象描述更多的实体。比如实体集合[(William Shakespeare),(Michael Jackson),(Barack Obama)]的覆盖率高的概念就包括是famous person, man, person，覆盖率逐步提高。

易见，典型性和覆盖率是相互矛盾的，我们需要有一个折衷的方案能够权衡利弊。如果我们的目标是更抽象地描述一个关系，那么对writer这个例子来说，[(work, person)]是集合最小、典型性适中、覆盖率较高的一个概念对集合。如果我们的目标是更准确地抽象描述一个关系，那么[(song, artist), (firm, director), (book, writer)]是集合较小、典型性高、覆盖率次高的一个概念对集合。常见的一个折衷方案是：在保证覆盖率的情况下，概念的典型性尽可能高。如[(William Shakespeare),(Michael Jackson),(Barack Obama)]的三个全覆盖的概念famous person, man, person中，famous person较其它两个就更典型一些。

实际上，以上所讨论的问题都是实体概念化（entity conceptualization）的所研究的问题。实体概念化的目标是由一个实体集合生成一个概念，是(n:1)的。而对实体对概念化来说，由3.1.1节，我们可以将问题大致分为三种目标：

1. 如果目标是由n个实体对生成1个概念对，那么生成的概念对将依赖isA关系进行关系分类。比如，关系r1等价于概念对(c11,c12)，关系r2等价于概念对(c21,c22)，如果概念间满足c11≤[[7]](#footnote-7)c21且c12 ≤c22，那么(c11,c12)≤(c21,c22)，即r1是r2的一个子关系。这个目标所要求的就是常见的折衷方案，即在保证覆盖率最高的情况下，概念的典型性尽可能高。
2. 如果目标是由n各实体对生成m个概念对，那么生成的概念对存在子关系的意义，因而能天然地建立关系分类体系，同时从m个概念对中也可以发现新关系。这个目标所要求的折衷方案有所不同，因为知识图谱的数据集存在偏移[[8]](#footnote-8)（bias）和噪音，满足覆盖率高的m个概念对可能会存在偏移和干扰项。偏移是由于数据不平衡或者不全所造成的，而噪音是一些无效的概念对，我们会在3.1.3节中详细解释。噪音可以通过设定阈值来过滤，而偏移本质上是信号量不足导致而难以控制。为了让关系分类体系的置信度更高，生成的m个概念对应该尽可能准确，因而这个目标的折衷方案应该与目标一的方案相反，即关系对概念化生成的概念对应该优先考虑高典型性，而覆盖率次之。
3. 如果目标是建立完整的关系分类体系，那么以上两个目标的结果结合在一起就可以建立关系分类体系，同时可以进行关系自动补全。

这三个目标中，第二个目标是最关键的，由第二个目标可以推导出第一个，进而推导出第三个目标。

因此，**本文研究的目标是：输入n个*实体对*，输出m个*概念对*，原则上优先考虑高典型性的*概念对***。

* + 1. 问题的难点

明确了目标之后，我们进一步分析目标的难点。与实体概念化不同，实体对概念化存在以下几个区别：

1. **概念对是未知的**。在实体概念化中，概念化的结果是已知的实体，而概念对不存在于知识图谱中[[9]](#footnote-9)，因此我们需要生成所有可能的候选概念对（candidate concept pairs），再从所有候选概念对中选择概念化较典型的几组。
2. **概念对的典型性**。一个典型的概念对不等价于一对典型概念的组合。依然举writer的例子，如果分开对左右两边的实体进行n对m的概念化，则左边生成的典型概念有[song, firm, book]，右边生成的典型概念有[artist, director, writer]。如果随意组合左右两边的典型概念，显然会出现(song, director)这个不合适的概念对。
3. **数据偏移**。偏移实是因为知识图谱不完全所导致的。具体地说，知识图谱中存在两类偏移。**第一类偏移**是指某个关系r的*实体对集合*E(r)中，可能存在大量的某种子关系的*实体对*，而其它子关系的实体对数目相对很少。比如writer这个关系在DBpedia中绝大部分实体对都是关于音乐创作者和电影创作者的，而很少出现书作者。很明显，这样的偏移会使抽象过程中数量多的实体对成为主导，从而“排挤”小的潜在关系。**第二类偏移**是指由于知识图谱中isA关系不全。比如family这个关系中，有一部分实体有butterfly这个概念的isA边却没有insect这个概念的isA边，因而这些实体对对一部分与其相关的概念对没有产生贡献。此外，当这种实体有一定数量之后，会使某个局部的概念对孤立出现，而与全局的概念对缺少关联。
4. **噪音。**由于概念对是未知的，在生成概念对的过程中必定会产生许多与实体关系无效的概念对，成为噪音。抽象的层次越高，噪音的影响越大。所以要有一定的剪枝、过滤方式降噪。
   1. 算法设计

实际上，如3.1.1所述，对某个关系的*实体对概念化*是一个逐步抽象的过程，运用这个思想，本文提出了一个两步抽象自下而上的算法。第一步通过抽象实体对生成候选概念对，第二步对通过聚类候选概念对并选择代表概念对而生成描述实体关系的一组概念对，这组概念对是可对Top-K进行排序的。

|  |
| --- |
| **算法框架** 两步抽象自下而上概念化方法 |
| **输入**：知识图谱G，实体关系r  **输出**：一组概念对CP(r)   1. 概念对生成   **输入：**知识图谱中关系r的所有实体对E(r)  **输出：**Top-K候选概念对   * 1. 由概念分类体系生成概念对空间C×C   2. 由E(r)为概念对进行排序  1. 聚类压缩   **输入：**候选概念对  **输出：**一组概念对CP(r)   * 1. 生成以概念对为点，相似度为边权的无向有权图   2. 马尔科夫聚类算法聚类   3. 选择簇和代表概念对 |

算法 1 算法框架

在分析推理的过程中，我们坚持将一对实体或一对概念当做一个对象来考虑。因为实体关系实际上是存在于一对实体/概念中的，是一对实体/概念间的潜在语义关系。在抽象的过程中，我们必须要保持实体对/概念对的完整性，弱化实体/概念的独立性，才能得到对实体关系最好的抽象描述。

* + 1. 概念对生成

给定一个关系r和知识图谱G，首先我们需要生成所有概念对，然后从中选择候选概念对，最后根据典型性将候选概念对进行排名。

首先，对知识图谱G中所有概念的集合C进行笛卡尔积计算，生成所有概念对集合C×C，再检索边集合E(G)得到具有关系r的集合E(r)：

接下来，我们将计算每一个候选概念对的典型性，然后根据典型性对候选概念对进行排序。如图3所示，C×C其实构成了一个二分图，图中的每个点代表一个概念，边(ci,cj)代表一个概念对，边权值代表了一个概念对的典型性。初始化的时候边权值都为零，然后遍历E(r)中的每一对(ei,ej)，通过计算一个典型性函数为不同的边(ci,cj)增加权值，遍历结束后所有边将有一个累积权值（cumulative weight）。形象地说，每一对实体对(ei,ej)都有“投票权”，对“偏好”的概念对(ci,cj)进行“投票”，当所有实体对投票结束之后，每个概念对就得到一个“得票总分数”。

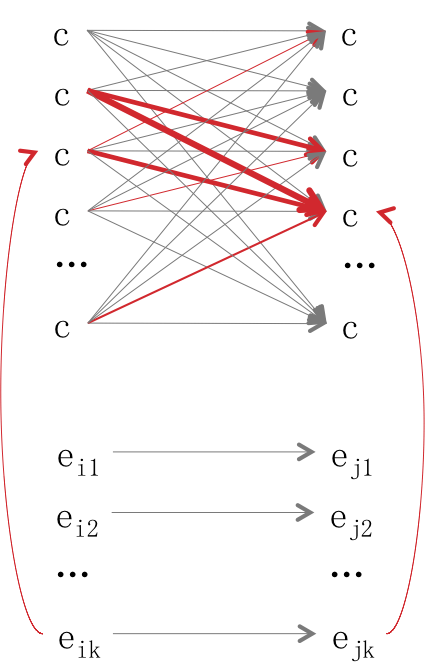
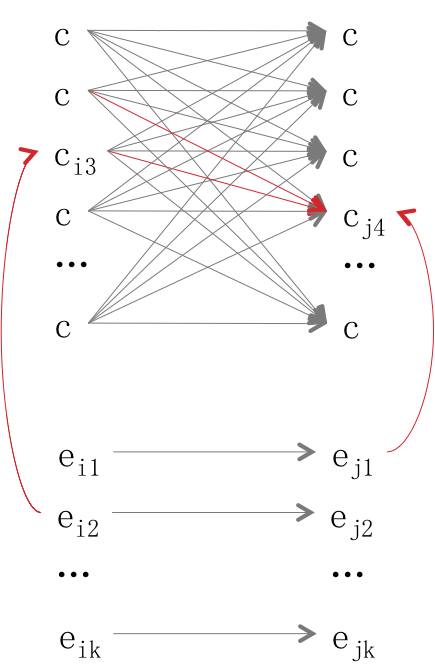
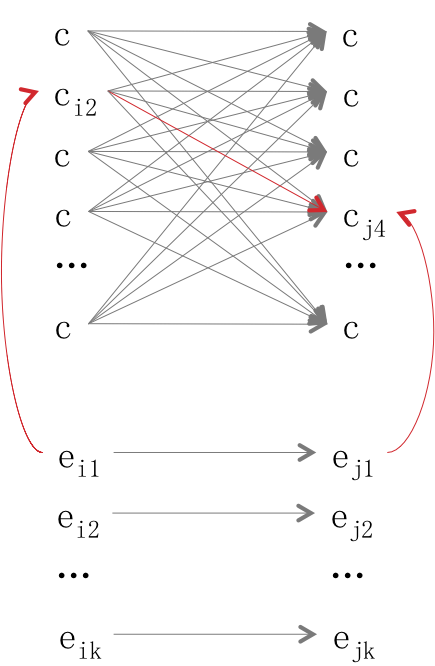


图 3 边累积权值迭代过程

需要注意的是，这种*概念对*的概念化方法与对左右两边实体分别概念化的方法有本质区别：前者是**边累积权值，**后者是**点累积权值**。**点累积权值**的方法的问题在3.1.3节中概念对的典型性已经说明。分别对两边实体概念化的方法会产生不符合实际情况的*概念对*，原因是我们的概念化目标是n:m，分别对两边实体抽象的m1个左概念和m2个右概念实际上组成了m1×m2个概念对，而这其中有许多组合是不正确在的。而**边累积权值**在每一次加权的时候虽然仍然会产生一些不正确概念组合（概念对），但随着遍历E(r)的累积，全局的边权值的累加会让正确的概念对凸显出来，不正确的概念对的边权值不会得到很多的累加。这个道理就像政治大选一样，一个投票站的选票可能是投给各种各样人物的，而所有投票站的投票总和就能呈现出一种总体趋势或者分布。实际上，E(r)是对概念对的一种语义限制，这种限制保持了“一对”之间潜在的语义关系，而且随着|E(r)|的增长，这种限制变得更加严格，所保留的语义关系也就更加充实，进而使得概念对的分布变得更加明显。

遍历后边累积权值的计算公式为：



需要讨论的是，典型性函数f的选择可以有多种方式，不同的方式产生的结果有一定差异。

* **平均累积（ACW，Average Cumulative Weight）**

定义典型性函数:



这种定义方式假设每一对实体生成的概念对都是平等的。这是一种最简单的定义方式，其典型性完全依赖E(r)的加和来实现，因而其累积结果的效果依赖于E(r)的大小，如果E(r)太小，其累积权值较大的边往往是比较“泛”的概念对而不是典型的概念对。

* **频数累积（FCW，Frequency Cumulative Weight）**

定义典型性函数:



这种定义方式的理由是，知识图谱中记录的isA关系附带有一个频数n(ei,ci)，这个频数是在知识图谱自动构建过程中ei和ci在语料中共同出现的次数。这种定义的好处是，对一些出现次数多的ei和ci，我们假设这样的实体ei有更高的可靠性抽象成为概念ci。然而，这样的假设只适用于一个小的E(r)。当E(r)的模比较大时，两个频数的乘机会因为某些特别大的频数把另一个小频数的效果放大，因而许多噪音也被放大了，一个“写了许多歌词的书作者”会把writer中的(book, artist)或者(song, author)放大，从而影响全局的分布。而对于较小的E(r)，可能其抽象化的目的就是表达某种特殊的意义，所以不能说某种概念组合是噪音。

* **信息量累积（ICW，Information Cumulative Weight，图3）**

以上两种方法都是非常直观的定义，接下来我们给出一种从信息量推导出来的典型性函数。互信息[28]（MI，Mutual Information）是信息论中的一种信息量，其表示一个随机变量由于已知另一个随机变量而减少的不确定性。在这里，我们可以将概念与实体作为两个随机变量，那么关系的实体对就是已知的，因此我们可以计算概念由于关系的实体已知而减少的不确定性。对于一个概念对和一个实体对，我们可以利用点互信息（PMI，Pointwise Mutual Information）来计算已知实体对对概念对的支持度。单个概念与实体的点互信息定义如下：



对于概念对和实体对，我们同样考虑成两个随机变量(Ci,Cj)和(Ei,Ej)。每确定Ei的一个取值，Ci不确定性就会减少，同理再确定一个Ei的取值，Cj的不确定性就会减少。这种不确定性是信息量，因此每确定一个(Ei,Ej)的取值，(Ci,Cj)的不确定性的减少量是两个随机变量不确定性减少量的和，因此我们扩展到定义概念对和实体对的点互信息：



最后我们对整个E(r)中的(ei,ej)求和：



使用互信息的好处是1）给定一个实体对(ei,ej)，一个概念对(ci,cj)出现的概率比起不给(ei,ej)时(ci,cj)出现的概率高多少，表示了这个(ei,ej)对(ci,cj)有多大的支持度；2）信息量可以相加，因此在定义“一对”的信息量时可以简单将两个部分的信息量加和。

综合以上的考虑，我们的算法使用ICW作为典型性函数的定义。最终的ICW边权值函数我们考虑了标准化因此Z：



其中Z的计算为，



最终，每个概念对都得到一个累积权值，根据这个权值我们能为概念对进行排序[[10]](#footnote-10)，并选择Top-K个概念对作为候选概念对的输出。

例 2. *图4中有表示关系writer的候选概念对的ICW得分情况。可见，得分最高的概念对*(song,artist)*显然不能完全代表writer关系，因此一个关系实际上应该是包含多个概念对的。而得分最高的前几个概念对实际上包含了一些冗余的概念对，3.3.2节将继续分析并解决这个问题。*

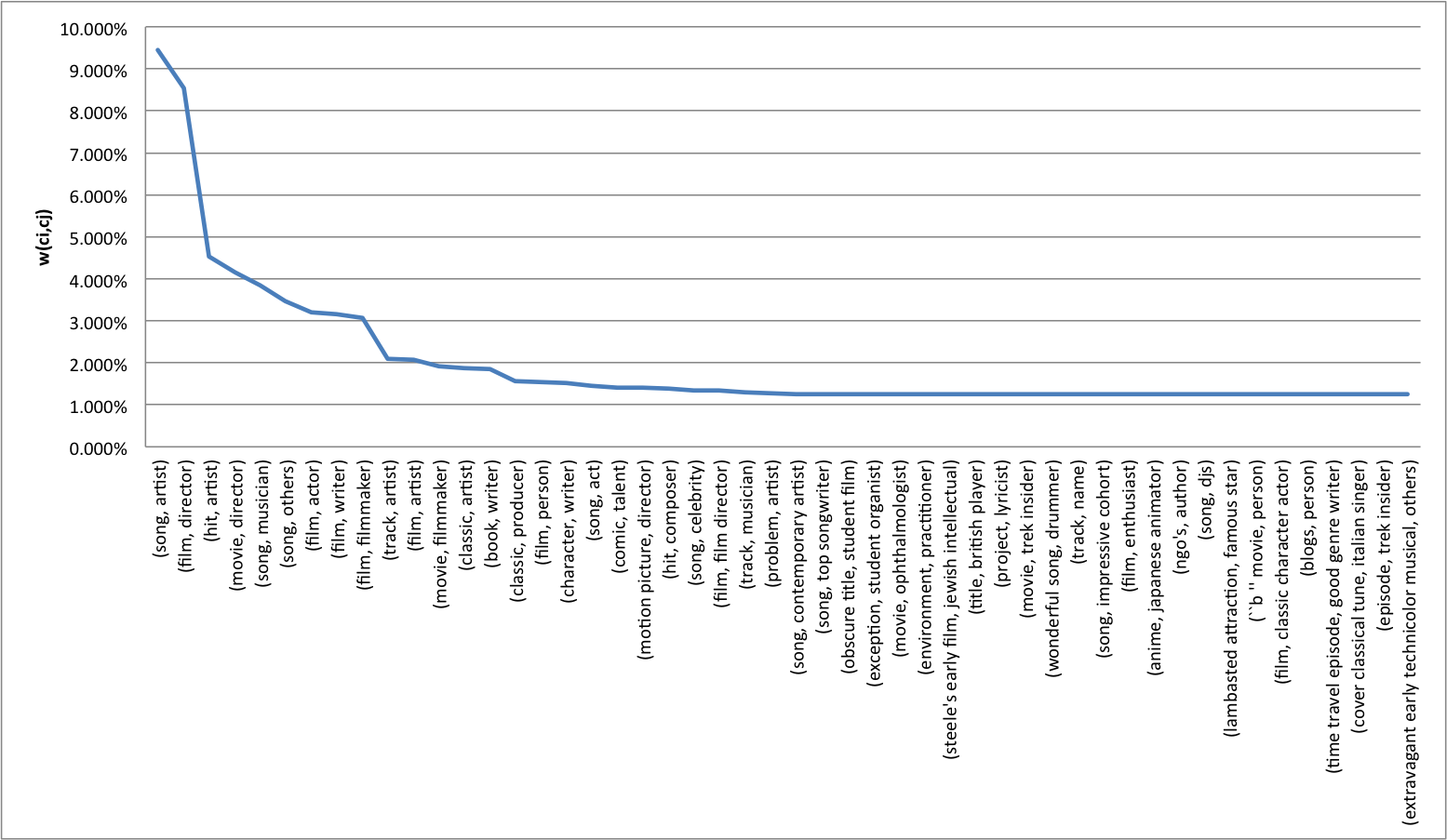


图 4 writer关系概念对ICW累积权值分布

* + 1. 聚类压缩

定义筛选后的*候选典型概念对集合*为C(r)，在3.1.2节中根据我们对问题的定义，我们希望得到一个多样化的*概念对集合*用于描述一个关系，从而能够根据多样化对我们的概念对排序进行修正。因此，我们需要第二步算法发现C(r中不同的类别。

此时的C(r)中有很多冗余概念对，比如writer这个关系在经过第一步概念对排序之后，[(song, musician),(song, artist),(firm, firm maker),(firm, director)]都是具有较高典型性的概念对，而其中明显存在一些相似度高冗余项，并且有明显的分类。因此，我们需要做的工作是：1）找到一种恒量概念对相似度的方法，2）通过这种相似度方法将不同类别的概念对聚合。

* + - 1. 概念对相似度

在考虑恒量概念对的相似度的方法时，我们需要运用之前给边投票的思想，即要把一对概念当做一个对象来考虑，而不要拆开考虑。因为一旦拆开考虑，就会损失概念对间存在的潜在语义关系，而这个潜在语义关系正是我们需要抽象描述的实体关系。

在3.1.2中，我们从实体概念化联想到了实体对概念化，从而提出了后续的思路。同样地，在这里我们需要借鉴对单个概念之间相似度流行的计算方法。在没有语料的情况下，计算两个概念的相似度可以通过WordNet来计算公共祖先，可以通过Word2Vec来计算语义相似度，也可以通过共享实体数来计算潜在语义，进而计算相似度。对于*概念对*来说，像WordNet这样的分类体系实际上是我们最终的目标；Word2Vec是一种词嵌入（Word Embedding），需要训练文本用机器学习的方法，而我们并没有概念对的训练样本，所以难以将*概念对*嵌入成一个向量；共享实体对成为了一个很有希望的方法，因为我们依然可以利用isA关系得到一个*概念对*拥有的*实体对*。所以，我们将使用共享实体对来计算两个概念对的相似度。

由于这里的概念对已经有了一定的典型性，所以在计算共享实体对时我们不考虑权重问题。此外，我们假设一个在知识图谱中比较“泛”的概念对，在一个子集E(r)中也是比较“泛”的。因此，我们定义在关系r下，与一个概念对(ci,cj)对应的一个实体对集合EPr(ci,cj)：



我们定义在关系r下，两对概念对(ci,cj)和(ck,cl)之间的相似度为对应实体对集合的Jaccard距离：



* + - 1. MCL聚类（马尔科夫聚类算法）

我们不妨创造性地把C(r)考虑成一个无向图，每个结点是一个概念对，节点的权值是ICW值，每条边表示概念对之间的相似度，边权值是3.2.2.1中所定义的Jaccard距离。由图5中我们看到，知识图谱中的信息对于这个关系有明显的第一类偏移，音乐和电影明显占抽象过程中的的主导地位。同时图6中，如果筛选阈值设定的不够合理，则会看到第二类偏移和很多噪音。

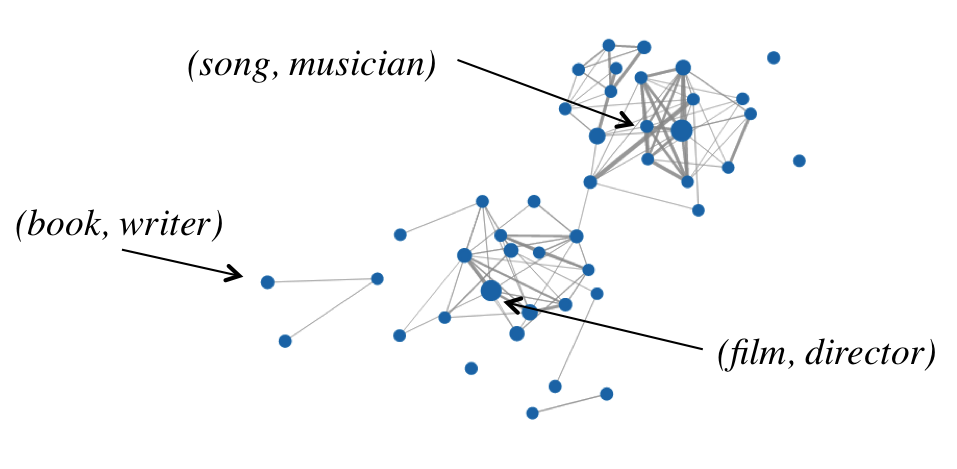


图 5 writer关系聚类图

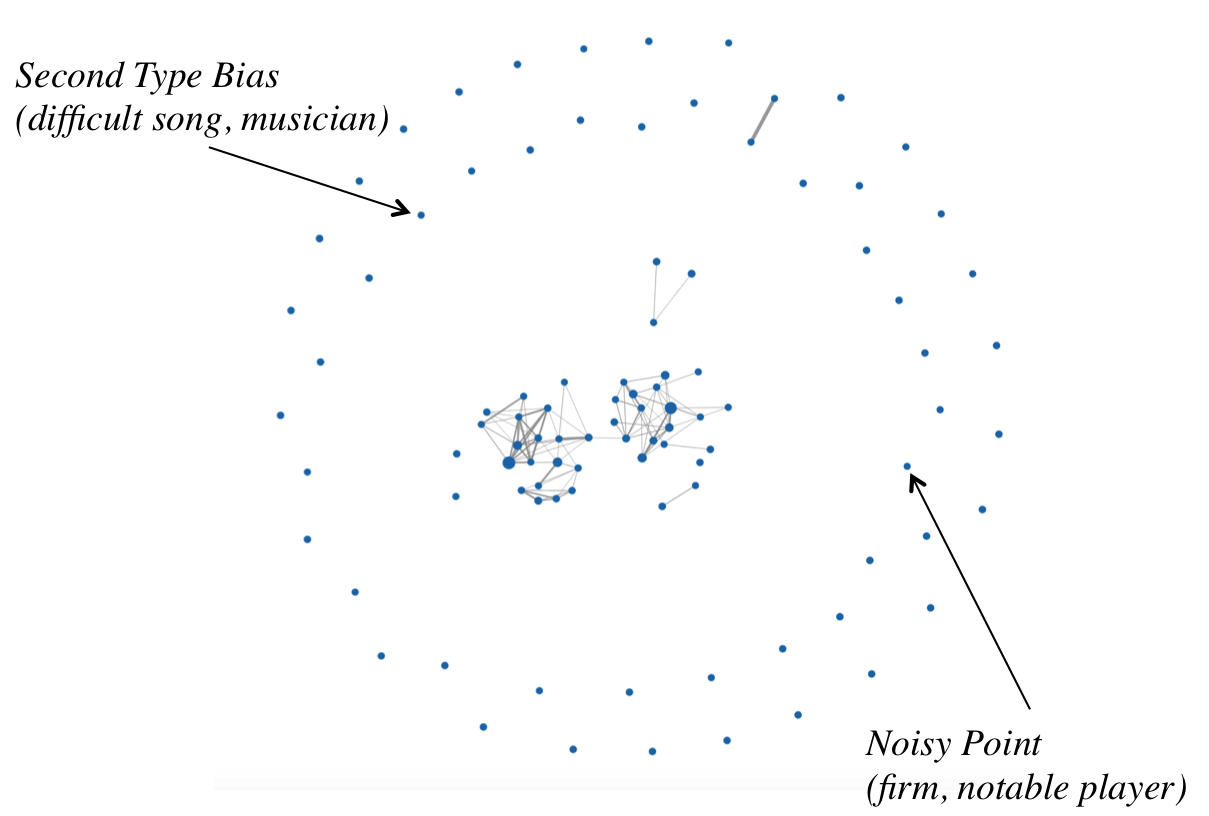


图 6 writer关系在阈值较大时的聚类图

显然，我们应该选择一种聚类方法并且具有以下的特点：

* 无向有边权值的稀疏图
* 簇的个数事先未知
* 能够适应各种形状
* 对噪音有很强的鲁棒性

因此我们选择马尔科夫聚类算法（MCL，Markov Clustering Algorithm，Dongen 2000 [22, 23]），该算法主要运用概率图上的随机游走思想，将矩阵迭代相乘至收敛，最后根据连通度分割簇。算法如下：

|  |
| --- |
| **算法1** 马尔科夫聚类算法 |
| **输入**: 无向图G，幂参数e，膨胀系数r   1. 根据G生成概率邻接矩阵M 2. 为M中每个结点增加自循环（可选） 3. 规范化矩阵M 4. 以e介幂扩展（Expand）矩阵：   M:=(M)e   1. 根据膨胀系数r对矩阵按列膨胀（Inflation）：   foreach column i of M  M(:,i):= (M(:,i))r  M(:,i):= M(:,i)/Sum(M(:,i))   1. 重复4、5两步直到矩阵收敛 2. 根据连通度切分矩阵获得簇 |

其中，第5步的意思是随机点在概率图上随机游走e步，通常e=2，即每一次循环只随机游走一步。第6步是放大每一个结点的出边权值（和为一）的差异，即原来的概率p变为pr，归一化后使概率大的边变得更大，概率小的边变得更小。膨胀系数r大则矩阵收敛得快，簇更多；膨胀系数小则矩阵收敛慢，簇更少。

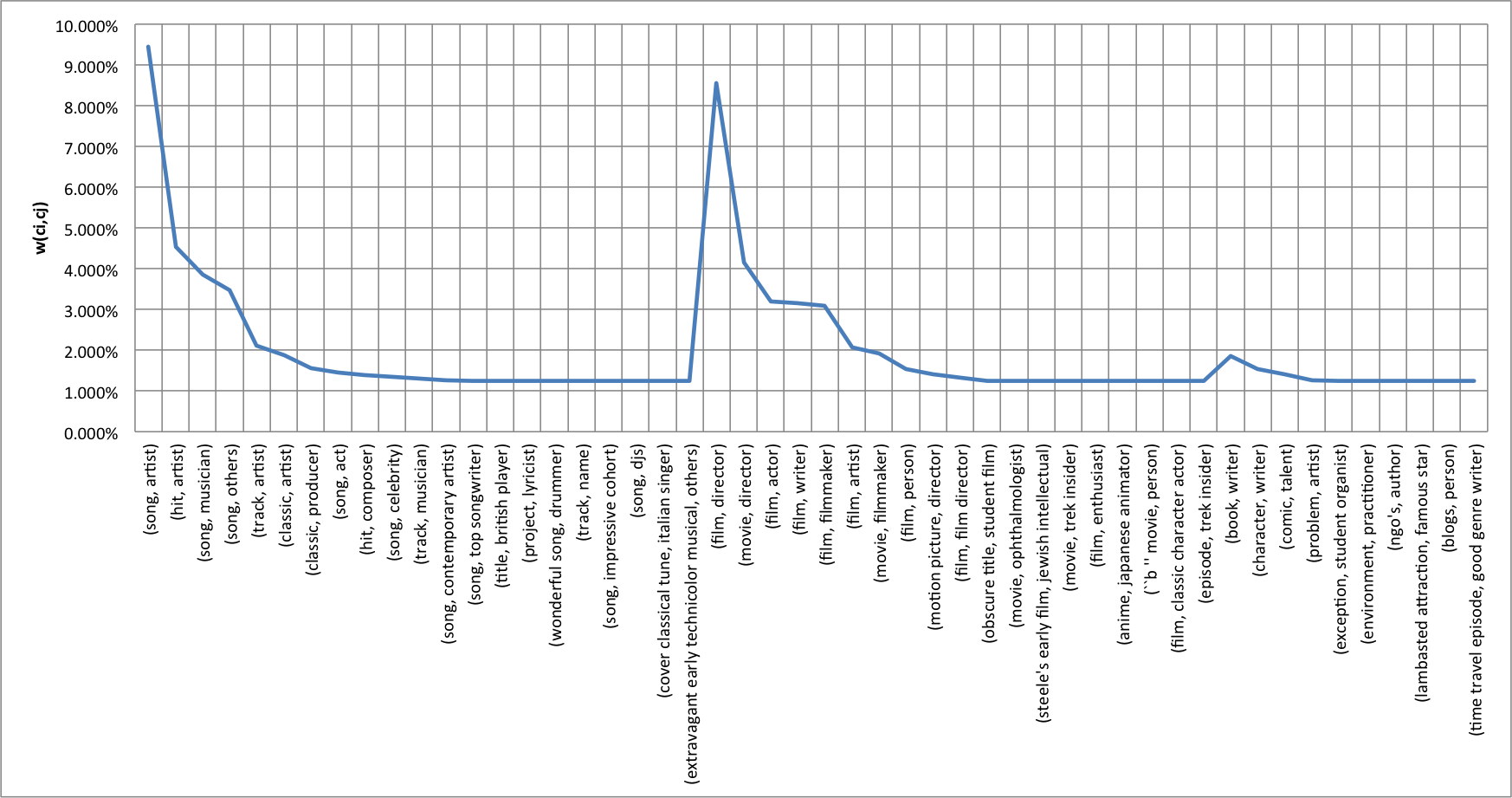


图 7 聚类后writer关系概念对ICW累积权值分布

*例 3. 图7表示关系writer的候选概念对在聚类之后的ICW分数分布。有三个簇明显可见，每个簇中得分最高的概念对分别是*(song, artist)*,* (film, director)*和*(book, writer)*。*

通过MCL聚类，C(r)中的概念对被分为了多个簇，每个簇代表了一类概念对，簇内的所有*概念对*具有潜在的共同语义，这个潜在的共同语义抽象就是对一群实体对的抽象，即实体关系。

* + - 1. 簇的选择与代表概念对

我们定义通过MCL聚类产生的簇为C(r)的一种划分CL(r)，最后我们要将划分中的噪音去除，然后根据不同概念对的典型性和覆盖性进行排序。

MCL算法对噪音是鲁棒的。根据3.1.3的噪音分析，*干扰概念对*与其他概念对的区别是其相对独立，同时我们采用的Jaccard距离概念对间不是句法上的相似度，而是潜在语义的相似度。区别于一些基于簇中心距离的算法（如K-means），MCL聚类没有簇中心，其随机游走保证了相对独立的子图间不能产生流动，所以噪音无法影响别的簇，这也天然地符合了我们的原则，即主要考虑典型性次要考虑覆盖率。因此，CL(r)中的许多噪音都被聚为小型的独立簇，这些簇不具有典型性，因此将其从CL(r)中去除去。

去除噪音后，我们还要考虑簇的质量问题。一个最直观的方法是，一个簇所包含的实体对越多，说明支持这个簇的信号越多，那么簇的质量或者可靠性就应该越高。因此我们可以按照簇的大小来选择有效的簇。在实验部分我们将根据不同簇大小进行聚类效果的评估。

经过以上两步筛选后，现在我们需要考虑最终由簇生成的概念对的排名策略。一个最简单的方式就是找出每个簇中最典型的概念对，即ICW最大的概念对，然后对每个簇根据簇大小进行遍历，生成|CL(r)|个概念对，每个概念对对应一个簇。根据簇的大小进行遍历，一是考虑了概念对的多样性，二是考虑到越大的簇通常包含更多的信号，因而排名也应该越高。

但是，最典型的概念对往往不能代表整个簇，因此我们除了ICW以外应该考虑一个簇内概念对之间的覆盖程度。这里我们借鉴PATTY系统中的弱包含关系和威尔逊分数，来恒量*概念对*S对*概念对*B的支持度，即簇中概念对之间的覆盖程度。这种计算方法实际上是最大似然估计，通过概念对S中的实体对样本去估计真实的概念对下所有实体对对B的支持度。然而如果S集合的元素太少，这种最大似然估的方式未必有较高的置信度。因此，我们需要对这种弱包含关系进行修正，即需要考虑集合样本的大小问题。注意到“一个*概念对*S下的*实体对*是否属于*概念对*B下的*实体对*”实际上是一个伯努利分布，所以我们采用威尔逊分数用于后验估计真实的集合支持度。根据威尔逊区间的定义，我们需要考虑置信区间的计算方法，通常在存在小样本的时候，我们不使用正态分布区间，而是使用Edwin Bidwell Wilson的修正公式：



其中表示对某个置信水平的统计量。一般情况下，在95%的置信水平下，z统计量的值为1.96。

因此，我们可以为簇cl中的每个概念对定义一个质量指标：



根据这个质量指标，我们可以从每个簇中选择一个代表性的概念对，这样如果一个关系r有N个簇，那么我们就可以生成N个概念对，并且根据簇的大小对概念对进行排序。为了扩展到前K（K>N）的情况，我们可以简单地使用轮询的方式生成K个概念对：每轮从最大的簇开始，取簇中s(ci,cj)最高的概念对，然后取次小的簇；当取完最小的簇之后，再从最大的簇开始新一轮，直到获得K个概念对为止。

回忆3.1.2节，其中中第一个目标的n:1抽象实际上是n:m抽象的一个特例，因此当CP(r)中只有一个簇时，簇的第一个中心代表实际上就是n:1的最终结果，这也印证了我们选择的目标是合理的。

1. 实验
   1. 数据处理

因为我们的算法设计是基于二元关系的，所以我们选择DBpedia中的对象属性（ObjectProperty）作为实体关系的数据集，每个对象属性在DBpedia中有大量的实体对。而概念分类体系，我们选择当前概念数量最多的Probase。

我们取了DBpedia的实体与Probase的实体的交集。由于Probase没有进行过实体关联，所以我们在这其中进行了一些人工关联，从而使得交集更大。最终我们得到约20.6M的实体交集。

在概念对生成时，如果使用一个关系下的所有实体对，那么复杂度将非常高，因而我们进行了随机采样，即对一个关系随机采样500对实体，这500对实体来自于DBpedia和Probase的实体交集。

* 1. 准确率

实验选择的实体关系来自于DBpedia的对象属性，而每个对象属性都有相应的域（domain）和范围（range）用于限定对象属性的实体类型，某些对象属性的域或范围为全局。如birthPlace这个对象属性的域和范围是(Person, Place)，knownFor域和范围是(Person, #*[[11]](#footnote-11)*)。因此，对于我们算法给出的一个关系的一组概念对中，任意一对概念(ci, cj)，理论上应该满足ci≼domain且cj≼range，这个偏序关系等价上位词和下位词的isA关系。

我们随机选择了30个对象属性来进行域和范围分析。每一组实体对经过两步算法之后都能压缩为一组概念对，一组中的每一个概念对对应第二步聚类算法中的一个簇。我们对一个关系r中每一簇的每一对概念(ci, cj)检查与(domain, range)的偏序关系:



我们计算了对簇中所有概念对的准确率：



除此之外我们还计算了对于每个簇中心(mi, mj)的准确率：



由于DBpedia的域和范围通常都比较抽象，往往是我们给出概念在上位词关系链上的祖先结点，考虑到Probase的isA关系链搜索复杂度非常高，我们同时使用了WordNet进行分析。此外我们对DBpedia中的域和范围词也进行了人工消歧和关联，如DBpedia中“组织”使用的是英文单词“Organisation”，而WordNet和Probase都是使用的“Organization”。

实验结果如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Relation** | **Domain** | **Range** | **Precision** | **Top-3** | **Top-1** |
| notableIdea | Person | # | 100.00% | 1/1 | 1/1 |
| influencedBy | Person | Person | 100.00% | 2/2 | 1/1 |
| deathPlace | Person | Place | 100.00% | 1/1 | 1/1 |
| knownFor | Person | # | 100.00% | 1/1 | 1/1 |
| leader | # | Person | 100.00% | 1/1 | 1/1 |
| president | # | Person | 100.00% | 2/2 | 1/1 |
| influenced | Person | Person | 97.53% | 2/2 | 1/1 |
| location | # | Place | 97.30% | 1/1 | 1/1 |
| birthPlace | Person | Place | 96.25% | 1/1 | 1/1 |
| nationality | Person | Country | 88.30% | 2/3 | 1/1 |
| artist | MusicalWork | Agent | 88.24% | 2/3 | 1/1 |
| country | # | Country | 87.91% | 1/1 | 1/1 |
| writer | Work | Person | 87.80% | 3/3 | 1/1 |
| product | Organisation | # | 78.95% | 1/1 | 1/1 |
| director | Film | Person | 78.21% | 1/1 | 1/1 |
| family | Species | Species | 76.47% | 2/3 | 1/1 |
| genre | # | Genre | 74.12% | 2/3 | 1/1 |
| author | Work | Person | 71.88% | 2/2 | 1/1 |
| album | # | Album | 68.42% | 2/2 | 1/1 |
| **Average** | **#** | **#** | **89.02%** | **92.98%** | **100%** |

表 4 对比DBpedia Domain-range实验

其中Top-K表示算法最终输出的前K个簇中心（有一些关系的簇个数少于K），x/y指y个簇中心中x个正确。

实验结果表明（表4），本文提出的算法发现的粒度更细的实体关系的准确率在所有生成概念对中达到89.02%，Top-3情况下达到92%，Top-1情况下到达100%。此外，由实验结果我们知道，聚类后的一个簇中仍然存在一些被放大或者不正确的概念对，这也印证了为什么我们需要选择合适的簇中心概念对。很明显，簇中心概念对的正确率有所提高，而最大簇的簇中心概念对正确率为100%，这是因为簇中心是最具有代表性的一对概念，而最大簇是所有实体对抽象过程中获得支持度最多、最具典型的概念对，因而在实际应用中，迭代构建关系分类系统的时候为了保证正确率，每次迭代如果产生了多个簇，应该选择最大簇作为新的子关系。

* 1. 聚类效果

第二步聚类算法的实质是发现同一关系下的不同子关系，因此我们设计实验用于评估聚类是否能够发现一组来自不同的关系的实体对所包含的概念对。

我们人工构建了37组合关系，每个组合关系中的实体对来自随机选择的连个或三个关系，如city\_place表示这个组合关系中包含来自city和来自place的实体对，writer\_spokenIn\_title包含来自writer、spokenIn和title的实体对。每个组合关系中不同来源的关系对数目也不同，随机挑选50/100/300对关系对用于模拟真实关系中的数据偏移。

在3.3.2.3节中我们提到，越大的簇因为其信号越多，因而越发可靠，质量越高。因此我们根据不同大小作为选择簇的标准而推荐出的组合关系rc的概念对集合，我们比较每个组合关系rc内的簇大小大于N的概念对与所有关系ra的概念对集合，如果相似度高，则返回该组合关系rc包含某个关系ra。因为我们已知rc包含的关系，所以可以据此计算聚类算法产生的概念对集合的准确率、召回率和F分数。表5表示在簇大小限制为6的情况下36组关系的实验情况，图8表示随着簇大小限制的提高，整体准确率提高，召回率下降，F-score在簇大小限制为6的情况下最高。

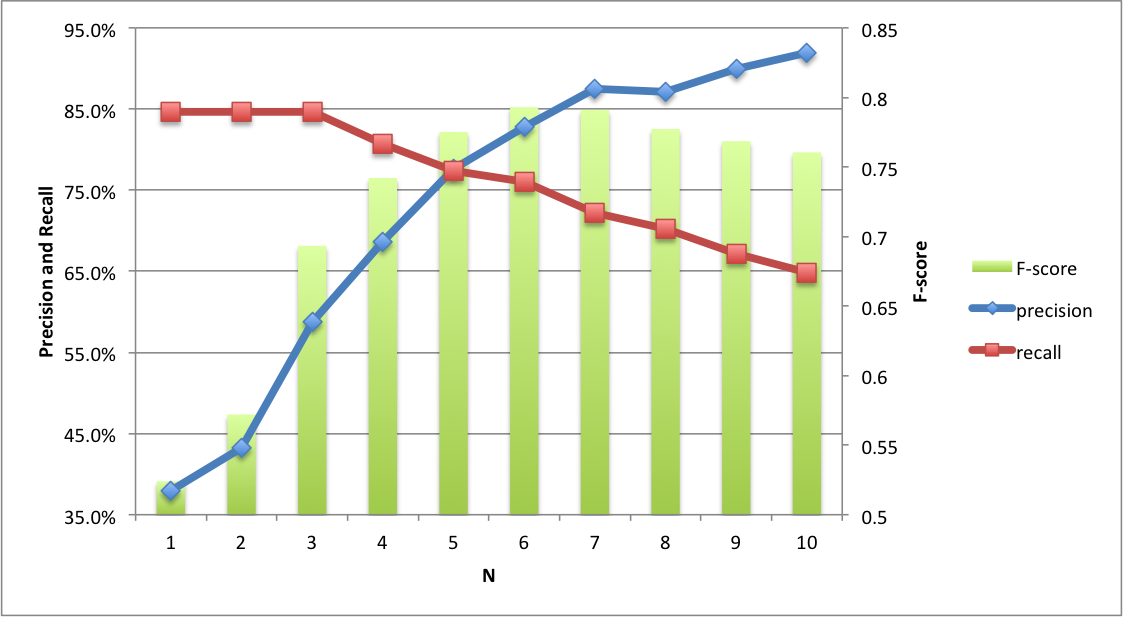


图 8 随着簇大小N变化的实验

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| combined relation | separate to | correct | number of true relations | number of found centers | precision | recall | F-score |
| city\_place\_ | [place, city] | 2 | 2 | 2 | 100.0% | 100.0% | 1.000 |
| influenced\_leader\_isPartOf\_ | [influenced, isPartOf] | 2 | 3 | 3 | 66.7% | 66.7% | 0.667 |
| author\_party\_party\_ | [author, party] | 2 | 2 | 3 | 66.7% | 100.0% | 0.800 |
| officialLanguage\_artist\_ | [officialLanguage, artist] | 2 | 2 | 2 | 100.0% | 100.0% | 1.000 |
| knownFor\_location\_ | [knownFor] | 1 | 2 | 1 | 100.0% | 50.0% | 0.667 |
| birthPlace\_album\_position\_ | [position, birthPlace] | 2 | 3 | 2 | 100.0% | 66.7% | 0.800 |
| director\_influenced\_family\_ | [influenced, director] | 2 | 3 | 2 | 100.0% | 66.7% | 0.800 |
| writer\_spokenIn\_title\_ | [title, spokenIn, writer] | 3 | 3 | 3 | 100.0% | 100.0% | 1.000 |
| isPartOf\_album\_notableIdea\_ | [isPartOf, notableIdea] | 2 | 3 | 3 | 66.7% | 66.7% | 0.667 |
| spokenIn\_mainInterest\_team\_ | [spokenIn, team, mainInterest] | 3 | 3 | 3 | 100.0% | 100.0% | 1.000 |
| album\_product\_ | [product, album] | 2 | 2 | 2 | 100.0% | 100.0% | 1.000 |
| nationality\_spokenIn\_knownFor\_ | [nationality, spokenIn] | 2 | 3 | 2 | 100.0% | 66.7% | 0.800 |
| genre\_notableIdea\_influencedBy\_ | [genre, influencedBy] | 2 | 3 | 3 | 66.7% | 66.7% | 0.667 |
| author\_director\_ | [director] | 1 | 2 | 1 | 100.0% | 50.0% | 0.667 |
| title\_party\_ | [title, party] | 2 | 2 | 3 | 66.7% | 100.0% | 0.800 |
| mainInterest\_deathPlace\_ | [mainInterest] | 1 | 2 | 1 | 100.0% | 50.0% | 0.667 |
| officialLanguage\_developer\_country\_ | [developer, country] | 2 | 3 | 3 | 66.7% | 66.7% | 0.667 |
| nationality\_title\_ | [nationality] | 1 | 2 | 1 | 100.0% | 50.0% | 0.667 |
| leader\_genre\_ | [genre] | 1 | 2 | 2 | 100.0% | 50.0% | 0.667 |
| city\_location\_ | [location] | 1 | 2 | 2 | 50.0% | 50.0% | 0.500 |
| author\_influencedBy\_ | [author, influencedBy] | 2 | 2 | 2 | 100.0% | 100.0% | 1.000 |
| country\_place\_ | [place, country] | 2 | 2 | 2 | 100.0% | 100.0% | 1.000 |
| officialLanguage\_position\_ | [position, officialLanguage] | 2 | 2 | 3 | 66.7% | 100.0% | 0.800 |
| position\_team\_ | [position, team] | 2 | 2 | 2 | 100.0% | 100.0% | 1.000 |
| team\_birthPlace\_place\_ | [team, place] | 2 | 3 | 2 | 100.0% | 66.7% | 0.800 |
| family\_director\_ | [family, director] | 2 | 2 | 3 | 66.7% | 100.0% | 0.800 |
| president\_nationality\_ | [president, nationality] | 2 | 2 | 2 | 100.0% | 100.0% | 1.000 |
| city\_country\_ | [country] | 1 | 2 | 1 | 100.0% | 50.0% | 0.667 |
| president\_president\_notableIdea\_ | [president] | 1 | 2 | 1 | 100.0% | 50.0% | 0.667 |
| product\_developer\_ | [developer] | 1 | 2 | 2 | 50.0% | 50.0% | 0.500 |
| deathPlace\_writer\_knownFor\_ | [writer] | 1 | 3 | 2 | 50.0% | 33.3% | 0.400 |
| ethnicGroup\_isPartOf\_location\_ | [location, isPartOf] | 2 | 3 | 3 | 66.7% | 66.7% | 0.667 |
| mainInterest\_developer\_family\_ | [developer, family, mainInterest] | 3 | 3 | 5 | 60.0% | 100.0% | 0.750 |
| birthPlace\_product\_genre\_ | [genre] | 1 | 3 | 1 | 100.0% | 33.3% | 0.500 |
| deathPlace\_influencedBy\_ | [influencedBy, deathPlace] | 2 | 2 | 2 | 100.0% | 100.0% | 1.000 |
| influenced\_leader\_ | [influenced] | 1 | 2 | 1 | 100.0% | 50.0% | 0.667 |

表 5 在簇大小限制为6的情况下36组关系的实验情况

* 1. 质量评估

算法的最终输出是一组Top-K簇的中心概念对。为了评估最终给出的概念对是否具有较高的质量，同时为了评估算法的准确率和召回率，我们随机选取了30组关系进行人工打分，并使用平均准确率（MAP，Mean Average Precision），平均质量和随机质量进行评估。

每个关系我们提供10组概念对，其中前几个由算法生成的概念对至多K对概念，余下的概念对是根据关系中的实体对生成的随机概念对[[12]](#footnote-12)。每个关系我们得到了10组打分（3分：相关，概括性高，2分：相关，有一定典型性，概括性适中，1分：相关，但太特殊，0分：不相关）。下表是其中writer关系的得分情况[[13]](#footnote-13)：

|  |  |  |
| --- | --- | --- |
| 关系 | 概念 | 平均分 |
| writer | \* ’s writer is a \* |  |
| 算法生成 | (song, artist) | 3 |
| (film, director) | 2.2 |
| (book, writer) | 3 |
| 随机生成 | (descriptive title, mccarthy-blacklisted u.s. writer) | 0.7 |
| (song, entertainer) | 1.4 |
| (scent, music icon) | 0.3 |
| (song, guest) | 0.5 |
| (day-to-day issue, film personality) | 0.2 |
| (song, lady musician) | 1.4 |
| (neo-realist film, director) | 1.4 |

表 6 writer关系打分表。表中前三个是算法生成的，后七个是随机生成的。实际问卷中所有概念对随机打散，打分者不知道哪些概念对是由算法生成的还是随机生成的。

下表是所有关系的打分情况：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Relation | MAP | Random MAP | Quality | Random Quality |
| notableIdea | 100% | 11% | 2.70 | 0.77 |
| influencedBy | 100% | 63% | 2.85 | 1.35 |
| influenced | 100% | 38% | 3.00 | 1.11 |
| country | 100% | 0% | 2.70 | 0.34 |
| birthPlace | 100% | 11% | 2.90 | 0.60 |
| deathPlace | 100% | 0% | 2.70 | 0.25 |
| party | 100% | 43% | 2.70 | 1.34 |
| director | 100% | 44% | 3.00 | 1.04 |
| writer | 100% | 0% | 2.73 | 0.84 |
| title | 100% | 0% | 2.23 | 0.64 |
| nationality | 67% | 0% | 2.70 | 0.69 |
| knownFor | 100% | 22% | 2.60 | 0.96 |
| officialLanguage | 100% | 0% | 3.00 | 0.47 |
| author | 50% | 25% | 1.50 | 1.13 |
| ethnicGroup | 100% | 0% | 2.30 | 0.54 |
| family | 100% | 29% | 2.33 | 0.98 |
| place | 100% | 11% | 2.40 | 0.90 |
| developer | 100% | 13% | 2.70 | 0.75 |
| genre | 67% | 14% | 2.37 | 0.77 |
| spokenIn | 100% | 11% | 2.70 | 0.80 |
| foundedBy | 100% | 11% | 2.80 | 0.82 |
| location | 100% | 0% | 2.70 | 0.79 |
| city | 67% | 0% | 1.83 | 0.66 |
| leader | 100% | 22% | 2.60 | 0.76 |
| team | 100% | 22% | 2.90 | 0.84 |
| president | 100% | 13% | 2.65 | 0.71 |
| product | 100% | 11% | 2.90 | 0.83 |
| position | 100% | 0% | 2.00 | 0.50 |
| artist | 33% | 14% | 1.47 | 0.93 |
| album | 50% | 0% | 1.90 | 0.59 |
| **Average** | **91%** | **14%** | **2.53** | **0.79** |

表 7 关系打分准确性与质量

MAP表示平均准确率，准确性等价于相关性，即由算法给出的概念对如果打分是好或者中的话我们记为相关，其它则不相关。Random MAP表示对随机生成的概念对的平均准确率。实验表明，算法生成的概念对的相关性准确率高达91%，远远超过随机生成概念对14%的准确率。Quality表示每个关系算法给出的概念对的平均打分，Random Quality表示随机生成的概念对的平均打分。实验表明，由算法生成的概念对的质量远远高于随机生成的质量。

|  |  |  |  |
| --- | --- | --- | --- |
| Method | MAP | Quality | |CP(r)| |
| Random | 14% | 0.79 | \* |
| ACW | 90% | 2.41 | 1.52 |
| ICW | **91%** | **2.53** | **1.96** |

表 8 基于概率的概念分类体系和非概率的比较

实验中，ICW权值累积我们使用了Probase提供的基于概率的概念分类系统，而常见的YAGO、Freebase都不是基于概念的，因此只能只用ACW权值累积。表8表示了这两种方法下人工打分的平均准确率、质量和平均发挥的概念对数目。

1. 结论
   1. 研究结论

在信息抽取中，实体关系抽取依赖于关系分类体系的构建，因而对实体关系的识别和特征描述是非常重要的。现有的关系识别和分类主要是基于本体的，针对同一对实体类型下分析关系模式句法而进行的，关系分类体系粒度较粗，此外对于新关系的发现也是局限于一对固定好的实体关系下的实体对聚类。因此，为了识别更细粒度的关系，我们需要有一种实体类型更丰富、基于语义的描述关系的方式。

本文应用概念分类体系，提出两步抽象自下而上的关系概念化方法，根据概念的典型性、多样性、覆盖率等特征为一个实体关系给出一组Top-K的概念对。本文提出的算法特点在于：1）使用实体类型更丰富的概念分类体系，从而能发现粒度更细的实体关系；2）将一对实体或一对概念当做一个对象来考虑，从而保留了一对实体或概念之间潜在的实体关系，通过大量实体对的叠加优化而选择更典型的概念对。算法的优势在于：不局限于一对粗粒度的实体类型，而是能够对实体对进行自下而上的抽象化而生成实体类型粒度更细的实体关系，有助于基于语义的关系分类体系的构建，也有助于发现更细的实体类型之间的新关系。

实验表明，本文提出的算法可以发现一个实体类型粒度更细的实体关系，也可以从一个组合实体关系中以较高的准确率分离关系，同时本算法生成的用于描述关系的概念对具有较高的质量。

* 1. 应用

给定一个关系，根据算法能获得关系的概念描述。这个方法可以有许多应用，包括前文所提及的关系分类体系构建，还有模板匹配消歧（Schema Matching Disambiguation）等涉及到需要关系特征的应用，根据不同的应用，关系概念化过程中选择的方法以及输出结果也会不同。

* + 1. 基于语义的关系分类体系

PATTY系统在构建关系分类体系时依赖的是句法特征。依赖于句法的关系分类体系在实际上不利于机器理解，因为理解过程实际上还只是简单的字符串比较而已。而基于语义的关系分类体系应该能够根据实体关系的语义来构建，比如本文中所提出的关系概念化就是一种实体关系的语义表达。关系分类体系的构建需要定义关系与关系之间的偏序关系，因此如果使用一组概念对描述一个关系的话，就需要定义一组概念对和一组概念对之间的偏序关系。一个简单的偏序关系就是描述概念支持度的弱包含关系。

基于语义的关系分类系统如果应用到关系抽取中，可以设计一些自举策略使得关系分类系统的搭建具有更高的准确率。比如writer这个关系，如果设定概念对的过滤阈值高的话，只能发现songwriter和filmwriter这两个子关系，而bookwriter由于偏移而无法出现。通过使用自举策略，系统可以再分裂子关系后同时继续抽取writer、songwriter、filmwriter的实体或者实体关系，当writer的实体对集合达到一定规模之后bookwriter就可以出现，而songwriter或者filmwriter达到一定规模以后还可以进一步分裂产生子关系。因此自举的方式不但可以使关系分类体系更加准确，其迭代的抽取方式还可以使得这种关系分类体系是自增的，这一点比起PATTY一次性构建关系分类体系来说是一个巨大的优点。

当然，如果需要构建关系分类体系，概念化的目标应该是覆盖度最大，因此在本文的基础上，应该采用迭代-收敛的策略，通过图的结构进行扩散（实体对到概念对，再由概念对到实体对，反复迭代），最终生成一组覆盖率最大、多样性最好的稳定的子关系概念对集合。

* + 1. 基于语义关系的模板匹配

现实中存在许多模板（Schema），包括数据库、电子表格和网页表格等。一个最基本的模板操作就是匹配：将两个模板作为输入，然后根据语义生成一个两个模板间元素的映射。表的第一列通常称为表头（Table Head），表头与其他列的是具有关系的，因此本文的算法可以根据表的两列元素内容（第一列表头和之后任意一列）生成两列关系的概念化描述，不同表的不同列如果具有相同或者相似的概念描述，就可以生成一种映射。根据模板匹配的分类，这种匹配方法可以归类为单匹配器方法（Individual matcher approaches）中基于内容（Instance/contents-based）的上下位词法。

在模板匹配中，关系的概念化实际上是一个特征，因此这组特征应该尽可能包含更多丰富的信息，比如最终生成的一组Top-K概念对是可排序的。因此，在这个应用中，最终生成的概念对可以采用轮询的方式选择出K个概念对作为一个模板的一列特征，然后进行匹配。

* 1. 改进方向
     1. 概率方法

现有的算法的设计实际上是启发式的，根据实验效果我们认为应该有概率方法用于解释。因此，一个改进的方向应该是设计一个优化目标，如，然后在概念对排序的时候考虑累积边权值的概率算法（现有的ICW分数是一种最简单的概率算法）；在聚类的时候也应该从优化目标出发来设计概率图模型来计算；在每个簇的代表概念对选择时也应该根据覆盖的概率模型来对每个簇生成数目不一的概念对（现有的典型性与覆盖率的乘积也是一种简单的概率算法）。

* + 1. 多元关系

本文提出的算法可以支持多元关系，如三元关系就可以将三元实体概念化为三元概念组，多元关系就可以将多元实体概念化为多元概念组。不过，因为多元关系在多元概念组生成的过程中会产生更高的复杂度，因此还需要考虑更多的加速方法。

* + 1. 偏移问题

偏移问题实际上可以在实际应用的过程中得到解决，比如设计一些策略用于判断是否应该将关系分裂为多个簇，或者在关系抽取中根据关系的实体对集合大小自举式地逐渐丰富关系的概念化。另外我们认为，从模型上也应该能通过大概念惩罚的方式解偏移问题，这也需要之后的工作继续探索。

1. 参考文献
2. Lee, Taesung, et al. "Attribute extraction and scoring: A probabilistic approach."*Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. IEEE, 2013.
3. de Abreu, Sandra Collovini, Tiago Luis Bonamigo, and Renata Vieira. "A review on Relation Extraction with an eye on Portuguese." *Journal of the Brazilian Computer Society* 19.4 (2013): 553-571.
4. Bach, Nguyen, and Sameer Badaskar. "A survey on relation extraction."*Language Technologies Institute, Carnegie Mellon University* (2007).
5. Schiff, Stacy. "Know it all." *The New Yorker* 31.07 (2006).
6. Rosario, Barbara, and Marti Hearst. "Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy." *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP-01)*. 2001.
7. Stephens, Matthew J., et al. "Detecting gene relations from Medline abstracts."*Pacific Symposium on Biocomputing*. Vol. 6. 2001.
8. Nastase, Vivi, and Stan Szpakowicz. "Exploring noun-modifier semantic relations." *Fifth international workshop on computational semantics (IWCS-5)*. 2003.
9. Kim, Su Nam, and Timothy Baldwin. "Automatic interpretation of noun compounds using WordNet similarity." *Natural Language Processing–IJCNLP 2005*. Springer Berlin Heidelberg, 2005. 945-956.
10. Nakov, Preslav, and Marti A. Hearst. "Solving Relational Similarity Problems Using the Web as a Corpus." *ACL*. 2008.
11. Nastase, Vivi, et al. "Learning noun-modifier semantic relations with corpus-based and WordNet-based features." *Proceedings of the National Conference on Artificial Intelligence*. Vol. 21. No. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
12. Turney, Peter D., and Michael L. Littman. "Corpus-based learning of analogies and semantic relations." *Machine Learning* 60.1-3 (2005): 251-278.
13. Yates, Alexander, et al. "Textrunner: open information extraction on the web."*Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics, 2007.
14. Fader, Anthony, Stephen Soderland, and Oren Etzioni. "Identifying relations for open information extraction." *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011.
15. Carlson, Andrew, et al. "Toward an Architecture for Never-Ending Language Learning." *AAAI*. Vol. 5. 2010.
16. Mohamed, Thahir P., Estevam R. Hruschka Jr, and Tom M. Mitchell. "Discovering relations between noun categories." *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011.
17. Carlson, Andrew, et al. "Coupling semi-supervised learning of categories and relations." *Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing*. Association for Computational Linguistics, 2009.
18. Wu, Wentao, et al. *Towards a probabilistic taxonomy of many concepts*. Technical Report MSR-TR-2011-25, Microsoft Research, 2011.
19. Hoffmann, Raphael, et al. "Knowledge-based weak supervision for information extraction of overlapping relations." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011.
20. Yao, Limin, et al. "Structured relation discovery using generative models."*Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011.
21. Nakashole, Ndapandula, Gerhard Weikum, and Fabian Suchanek. "PATTY: a taxonomy of relational patterns with semantic types." *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012.
22. Nakashole, Ndapandula, Tomasz Tylenda, and Gerhard Weikum. "Fine-grained Semantic Typing of Emerging Entities." *ACL (1)*. 2013.
23. Van Dongen, Stijn Marinus. "Graph clustering by flow simulation." (2001).
24. Van Dongen, Stijn. "A cluster algorithm for graphs." *Report-Information systems*10 (2000): 1-40.
25. Miller, George A. "WordNet: a lexical database for English." *Communications of the ACM* 38.11 (1995): 39-41.
26. Rahm, Erhard, and Philip A. Bernstein. "A survey of approaches to automatic schema matching." *the VLDB Journal* 10.4 (2001): 334-350.
27. Wu, Wentao, et al. "Probase: A probabilistic taxonomy for text understanding."*Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 2012.
28. Chu, Xu, et al. "KATARA: A Data Cleaning System Powered by Knowledge Bases and Crowdsourcing." *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015.
29. K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist*., 16(1):22–29, Mar. 1990.
30. 致谢

感谢我的导师，计算机科学与技术学院的肖仰华老师，在这几个月期间的认真指导。感谢韩国浦项科技大学（POSTECH）计算机系的Seungwon Hwang老师，在我的研究过程遇到瓶颈时提供建议。感谢计算机科学与技术学院的张义同学，在论文撰写期间全程为我提供帮助。感谢GDM实验室的各位同学，包括孙祥彦、谢晨昊同学等，在讨论中为我提供了有价值的资源。感谢我的母亲及其同学与同事、美国明尼苏达双城大学陈柯栋同学、复旦大学瞿佳庆同学、雷添羽同学、舒静同学、美国南加州大学马帅同学、云南师范大学计算机学院肖斓楠老师即其计算机学院的学生等，为我的实验进行了人工打分标注。

这篇论文将为我在复旦大学四年的学习与生活画上句号，因此我倾注了许多时间与精力。在接下来的几个月时间，我还将继续在肖仰华老师、Seungwon Hwang老师和张义同学的帮助下继续修改论文并投稿。

最后向审阅论文的各位专家和老师表示感谢，初次接触科研，论文中如有不足之处，敬请各位提出指正和建议。

1. 在信息抽取（IE）领域中，“实体”均指“命名实体”（Named Entity）。 [↑](#footnote-ref-1)
2. 百度百科《中文知识图谱》，复旦图数据管理实验室编制。 [↑](#footnote-ref-2)
3. “At codons 12, the occurence of point mutations from G to T were observed”翻译为“在第12号遗传密码子中观察到了从G到T的点突变”。 [↑](#footnote-ref-3)
4. http://rtw.ml.cmu.edu/rtw/ [↑](#footnote-ref-4)
5. d的计算根据隶属度概率的分布不同而不一。二项分布在样本集合较大时常采用正态区间估计置信度，而样本较小时就要依靠威尔逊区间来计算。 [↑](#footnote-ref-5)
6. 《The Angelic Conversation》，莎士比亚创作的十四行诗。 [↑](#footnote-ref-6)
7. ≤是偏序关系，这个偏序关系在具体构建关系分类系统的时候才能确定。 [↑](#footnote-ref-7)
8. 偏移，指知识图谱中的数据集不完全，不能涵盖现实生活中所有的实体或概念。3.1.3节中会详细描述。 [↑](#footnote-ref-8)
9. Domain-Range实际上是*类型对*，而不是*概念对*。 [↑](#footnote-ref-9)
10. 实际上，在实际算法编写的过程中，考虑到C×C的空间复杂度过高而且非常稀疏，因此实际上所有概念仅来自一个关系的所有实体的所有概念；遍历E(r)的时间复杂度也很高，因此实验中采用了随机采样。 [↑](#footnote-ref-10)
11. #表示全局范围 [↑](#footnote-ref-11)
12. 随机概念对要求不属于经过阈值筛选之后的CP(r)。 [↑](#footnote-ref-12)
13. 问卷链接：https://docs.google.com/spreadsheets/d/1Roj5bu376Lw7awmUNewWHZ0G4DJ7uLlQLnHc4vDaakw/edit?usp=sharing [↑](#footnote-ref-13)