An Approach to Automatically Constructing Domain Ontology [1]

Tingting He[1 2 3] Xiaopeng Zhang [1 3] Xinghuo Ye[1 3]

[1] Department of Computer Science, Huazhong Normal University
430079 Wuhan, China
[2] Software College of Tsinghua University 102201 Beijing, China
[3]National Language Resources Monitor & Research Center (network media)
430079 Wuhan, China

tthe@mail.ccnu.edu.cn zhangxiaopeng@mails.ccnu.edu.cn yxhez@tom.com

Abstract. In this paper we present an approach to mining domain-dependent ontologies using term extraction and relationship discovery technology. There are two main innovations in our approach. One is extracting terms using log-likelihood ratio, which is based on the contrastive probability of term occurrence in domain corpus and background corpus. The other is fusing together information from multiple knowledge sources as evidences for discovering particular semantic relationships among terms. In the experiment, we also improve the traditional k-mediods algorithm for multi-level clustering. We have applied our approach to produce an ontology for the domain of computer science and obtained promising results.

1 Introduction

Ontologies have become an important means for structuring knowledge and building knowledge-intensive systems. Ontologies have shown their usefulness in application areas such as intelligent information integration, information retrieval and natural language processing, to name but a few. For this purpose, efforts have been made to facilitate the ontology engineering process, in particular the acquisition of ontologies from domain texts [1].

Constructing an ontology is an extremely laborious effort. Even with some reuse of "core" knowledge from an Upper Model, the task of creating an ontology for a particular domain has a high cost, incurred for each new domain. Tools that could automate, or semi-automate, the construction of ontologies for different domains could dramatically reduce the knowledge creation cost. One approach to developing such tools is to rely on information implicit in collections of texts in a particular domain. If it were possible to automatically extract terms and their semantic relations from the text corpus, domain ontology could be built conveniently. This would be more cost-effective than having a human develop the ontology from scratch [2][3].

Domain specific ontologies are useful in systems involved with artificial reasoning, and information retrieval. Ontologies give such systems a vocabulary of terms, and concepts relating one term to another. In this paper, we present a method that automatically mining an ontology from any large corpus in a specific domain, to support data integration and information retrieval tasks. The induced ontology consists of domain concepts only related by parent-child links, not including more specialized relations. Our approach is comprised of two main phases: term extraction and relationship discovery. In the former phase, meaningful terms are extracted using LLR formula. In the latter one, parent-child relations among terms are induced through multilevel clustering.

The remainder of the paper is as follows. In Section 2, we will discuss the related works in this field. In Section 3, we will give an overview of the overall system architecture and explain the means and progress of ontology construction. Subsequently, an example will show some promising results we have

[1] The paper is supported by National Natural Science Foundation of China (NSFC), (Grant No.60496323; Grant No.60375016 Grant No.10071028;); Ministry of education of China, Research Project for Science and technology, (Grant No. 105117).

obtained when applying our mechanisms for mining ontologies from text, together with our analysis. This will be presented in Section 4. In Section 5, we conclude and mention our future work.

2 Previous Work

The existing approaches to ontology induction include those that start from structured data, merging ontologies or database schemas (Doan et al. 2002). Other approaches use natural language data, sometimes just by analyzing the corpus (Sanderson and Croft 1999), (Caraballo 1999) or by learning to expand WordNet with clusters of terms from a corpus, e.g., (Girju et al. 2003). Information extraction approaches that infer labeled relations either require substantial hand-created linguistic or domain knowledge, e.g., (Craven and Kumlien 1999) (Hull and Gomez 1993), or require human-annotated training data with relation information for each domain (Craven et al. 1998).

For automatic ontology construction, Govind and Chakravarthi have presented in 2001 an approach that extracts ontology from text documents by Singular Value Decomposition (SVD), which is a pure statistical analysis method , as compared to heuristic and rule based methods. They adopt Latent Semantic Indexing (LSI), which attempts to catch term-term statistical references by replacing the document space with lower dimensional concept space. Their method is convinced of its simplicity because it is based on fairly precise mathematical foundation. It is effective but limited with precision. What's more, it doesn't figure out the exact relations among terms [4].

In year of 2001, Dekang Lin and Patrick Pantel proposed a method for domain concepts discovery based on a clustering algorithm called CBC (Clustering by Committee). They generally regard a concept as a cluster of terms. It just deals with only one aspect of the whole progress of ontology induction [5].

3 Automatic Ontology Constructing

3.1 Architecture

An overall architecture for domain-independent ontology induction is shown in Figure 1. The documents of domain corpora are preprocessed to remove segments such as pictures and specific symbols and then filter the stopwords. Next, terms are extracted based on word segmentation and syntactic tagging. Subsequently semantic relations between pairs of terms are derived using multilevel clustering with evidences from multiple knowledge sources.

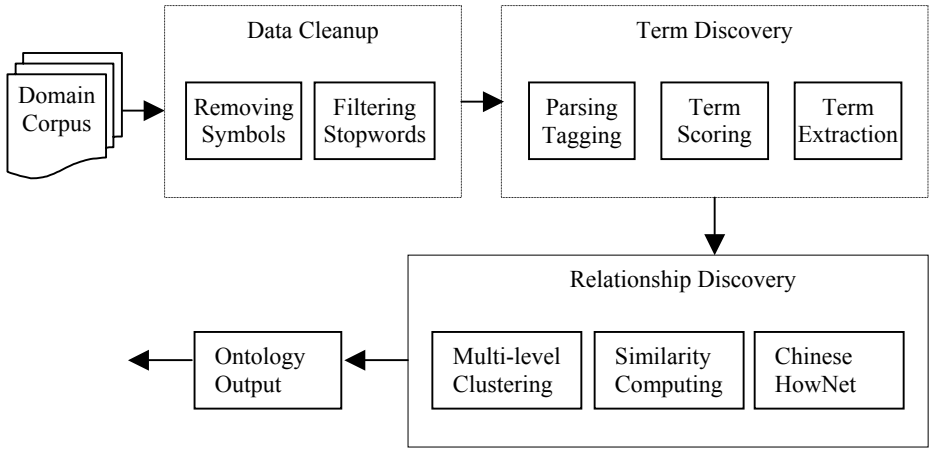

Fig. 1. System Architecture

3.2 Term Discovery

In this section, our goal is to identify domain-relevant terms from a collection of domain specific texts. For term extraction, there exist several approaches. One is based on PAT-TREE, which has the advantage in extracting terms (phrases) of any length because it could avoid word segmentation for Chinese. But it needs a large amount of documents to achieve excellent precision. Another approach to term extraction is C/NC-Value method proposed by Frantzi and Ananiadou (1999), which combines linguistics and statistics methods and makes a progress. But it has limitation because its linguistics knowledge is only for English.

In this paper, we propose a promising approach to extracting domain-relevant terms. Terms are scored for domain-relevance based on the assumption that if a term occurs significantly more in a domain corpus than in a background corpus, then the term is clearly domain relevant.

As an example, in Table 1 we compare the number of documents containing the terms "算法"(algorithm), "内存"(memory), "路由器"(Router) and "数据库"(database) in a domain-dependent corpus of computer science including 200 Chinese periodical papers, compared to a larger Modern Chinese Corpus as the background corpus.

Table 1. Term IDF in domain and background corpora

	算法	内存	路由器	数据库	*Total*
Domain Corpus	89	78	73	69	200
Background Corpus	7	0	4	6	12000
Difference	82	78	69	63	-11800

We can observe from Table 1 that the terms "算法"(algorithm), "内存"(memory), "路由器(Router) and "数据库"(database) occur significantly more in the domain corpus than in the Background corpus. We consider they are domain-dependent. To estimate the domain relevancy of a term, we use the log-likelihood ratio (LLR) given by

$$-2 \log 2 (Ho (p;k1,n1,k2,n2) / Ha (p1,p2;n1,k1,n2,k2)) \quad (1)$$

LLR measures the extent to which a hypothesized model of the distribution of cell counts, Ha, differs from the null hypothesis, Ho (namely, that the percentage of documents containing this term is the same in both corpora). We use a binomial model for Ho and Ha. Here, $p=(k1+k2)/(n1+n2)$, $p1=k1/n1$, $p2=k2/n2$, $k1$ is the number of documents containing the term in the domain corpus, $k2$ is the number of documents containing the term in the background corpus, $n1$ is the total number of documents in the domain corpus, $n2$ is the total number of documents in the background corpus [6].

3.3 Relationship Discovery

In this section, we try to mine the parent-child links among terms using multilevel clustering based on term similarity. We fuse together information from multiple knowledge sources as evidences for particular semantic relationships among terms. For clustering we adopt improved k-medoids algorithm, which is flat and effective.

The process of inducing relations is as follows. First the clustering algorithm is used to obtain top-level clusters. Subsequently, for each top-level cluster, we use the clustering algorithm again to gain clusters of second level. By analogy, we can find multilevel clusters that imply parent-child relationships among terms. In experiment the depth of clustering level is restricted.

3.3.1 Term Similarity Computation

The preprocessed documents are analyzed and the frequency matrix known as Term-Document matrix is produced as result of the analysis. For a Term-Document matrix M[m][n], each term can be represented as the following vector:

$$\vec{T}_i = (\text{M[i][0]},\ \text{M[i][1]}, \ldots, \text{M[i][k]}, \ldots, \text{M[i][n]}) \tag{2}$$

We can compute the cosine similarity between a pair of terms given by

$$\cos(\ \vec{T}_i, \vec{T}_j\ ) = \frac{\vec{T}_i \cdot \vec{T}_j}{\left|\vec{T}_i\right| \cdot \left|\vec{T}_j\right|} \tag{3}$$

The cosine similarities between several term pairs are shown in Table 2.

Table 2. Cosine similarities of term pairs

Term1	*Term2*	*Cosine Similarity*
缓冲区(buffer)	磁盘(disk)	0.436248
缓冲区(buffer)	主存(main store)	0.579771
内存(memory)	磁盘(disk)	0.434059
服务器(server)	网络(network)	0.17585

It can be observed that the cosine similarity is likely to satisfy the concurrent frequency of term pairs. It reflects the context correlation of terms but might have a prejudice against semantic correlation among terms. For instance, the cosine similarity computed above between "磁盘"(disk) and "数据" (data) is just 0.0663552, which deviates from the experiential value.

In order to exact similarity, we use HowNet as another knowledge resource, which is a Chinese thesaurus by Zhendong Dong and Qiang Dong (2001). We combine the similarity of terms in HowNet as a supplement with the cosine similarity to estimate the correlation of term pairs. The HowNet similarity that involves unknown words is assigned to 0. Table 3 shows some HowNet similarities of term pairs.

Table 3. HowNet similarities of term pairs

Term1	*Term2*	*HowNet Similarity*
缓冲区(buffer)	磁盘(disk)	0.149333
缓冲区(buffer)	主存(main store)	0.000000
内存(memory)	磁盘(disk)	0.149333
服务器(server)	网络(network)	0.285714

The final formula of similarity computation is given by

$$sim(\vec{T}_i, \vec{T}_j) = \frac{simA(\vec{T}_i, \vec{T}_j) + \alpha \cdot simB(\vec{T}_i, \vec{T}_j)}{2} \tag{4}$$

Where $simA(\vec{T}_i, \vec{T}_j)$ is the cosine similarity and $simB(\vec{T}_i, \vec{T}_j)$ is the HowNet similarity; Meanwhile, α is an adjustment argument. The adjusted similarity between term pairs is given in Table 4 (α =0.5).

Table 4. Adjusted similarities of term pairs

Term1	*Term2*	*Similarity*
缓冲区(buffer)	磁盘(disk)	0.510913
缓冲区(buffer)	主存(main store)	0.579771
内存(memory)	磁盘(disk)	0.508724
服务器(server)	网络(network)	0.318716

3.3.2 Clustering Algorithm

In order to make it converge faster and result in better clusters in accordance with the original distribution of terms, we improved the traditional k-medoids algorithm. When deciding the new center of a cluster, we first choose top-p terms closest to the old center in the cluster and take the term closest to the mean of the p terms as the new center. The improved algorithm is described as follows:

1. Choose m terms as the centers of clusters randomly: $C_1, C_2, \ldots, C_i, \ldots, C_m$;
2. Compute the similarity for each term to each cluster center. Assign the term into the closest cluster.
3. Determine the new center of each cluster as follows:

– Compute the average similarity of terms in cluster i using the formula:

$$avgsim[i] = \frac{1}{n}\sum_{k=1}^{n} sim(T_k, C_i) \tag{5}$$

Where n is the number of terms in cluster i.

– Pick out *p* terms most closest to the cluster i center decided by

$$p = m * \frac{avgsim\ [i]}{\max\_\ avgsim} \tag{6}$$

Where $\max\_\ avgsim$ is the maximum of m average similarities for m clusters.

– Compute the mean of the above *p* terms and choose the term closest to it as the new center of cluster i.

4. Go to step 2 only if m cluster centers is not stable.
5. End and result in m clusters.

4 Experiments and Evaluation

We have applied our approach to produce ontology in computer science domain. The domain corpus contains 200 Chinese theses (3.12 M) retrieved from professional publications. The background corpus we used is the Modern Chinese Corpus (1999-2001) which consists of 12000 documents (33.5 M).

4.1 Term Discovery

In experiment, we have compared the effect of the two scoring methods LLR and TF.IDF. Table 5 shows the scores of some terms in relative percentage. The boldfaced terms is identified by LLR to be domain-relevant. It can be inferred from the experiment results that LLR excels TF.IDF.

Table 5. Relative scores of both LLR and TF.IDF

Term	*Domain DF*	*Background DF*	*LLR score*	*TF.IDF score*
路由器	73	4	98.6	66.8
总线	56	1	99.9	70.4
控制器	35	7	96.5	94.5
端口	25	5	93.1	65.3
缓存	36	2	94.6	53.5
请求	17	188	67.5	72.6
线路	31	232	56.2	65.7
质量	37	3725	42.4	99.9
屏蔽	7	8	15.3	48.2

A term list of 286 domain-relevant terms (key terms) has been manually constructed to support the subsequent evaluation. We estimate the results normally based on Recall, Precision and F-measure, which are defined as follows.

- **Recall(R)**
  The ratio of correct terms number to key terms number in manual list given by

$$R = \frac{N_{correct}}{N_{key}} \tag{7}$$

- **Precision(P)**
  The ratio of correct terms number to extracted terms number given by

$$P = \frac{N_{correct}}{N_{correct} + N_{incorrect}} \tag{8}$$

- **F-measure(F)**
  F-measure is defined as a combination of recall and precision:

$$F = \frac{2RP}{R + P} \tag{9}$$

These three values vary depending on the threshold of LLR. Figure 2 shows the variety of F-measure, recall and precision along with LLR.

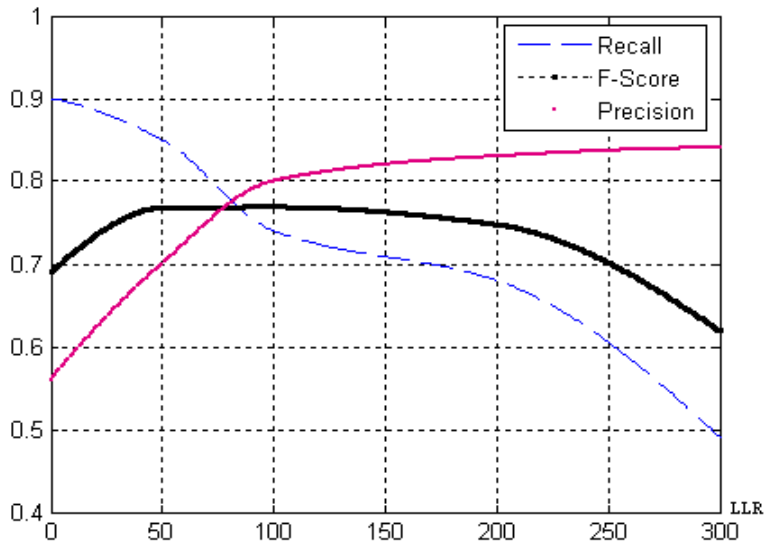

Fig. 1. F-measure, recall and precision by varying the threshold of LLR

4.2 Relationship Discovery

According to figure 2, we chose 94 as the threshold of LLR and obtained 216 domain-relevant terms. The depth of clustering level was restricted to 2. We have gained 5 top-level clusters and 15 second level clusters. The precision of the top-level clusters achieved 76.7%. The average precision of the second-level clusters achieved 70.3%. The parent-child relationship among terms and clusters is shown in figure 3. Each cluster is named by the most frequent term in it.

网络	网络	路由器 路径 主干网 速率 中继 广域网 延迟 分组 交换 交换机 局域网 交换网 接入网 策略 关键词 网格 网络 拓扑 骨干网
	搜索	搜索 引擎 网页 脚本 主机 万维网 网站 网址 带宽 阻塞 端口 监听 信息网
	服务器	防火墙 流量 计算中心 域名 主页 服务器 拷贝 电子邮件 邮件 邮箱 日志 数据流 语音 共享
	信息港	智能性 信息港 因特网 智能化 电话网 宽带 调制 解调 频率段 数字网
计算机	进程	体系 结构 总线 硬件 接口 中断 控制器 寄存器 运算 时间段 数据 共享 实时 系统 调度者 进程 子系统 分系统 硬盘
	计算机	计算机 芯片 大型机 巨型 磁盘 软盘 光盘 主存 吞吐 软驱 外存 内存 存储器 存储 缓冲 缓存 输出 驱动 驱动器 适配器 网卡
程序	参数	矩阵 数组 指针 字符串 参数 静态 访问 编译器 变量 字符 标量 流程图 流程 结构图 逻辑图 运算符
	编程	控制台 编程 源程序 程序包 子程序 程序性 源代码 向量 编译 调试 调试器 程序员 瓶颈 数据包
	地址	输入 提示符 键盘 终止符 示意图 主程序 操作符 空格 数据位 读取 存取 缓冲区 地址 计时器
	程序	编程者 查询 分析器 数据库 数据表 冗余 备份 标识符 计数器 初始化 程序 语句 表达式 操作数
	图形	切换 图形 界面 请求 文本框 图标 按钮 视图 标识号
	软件	软件 软件包 构建 复用 重构 组件 构件 重用性 模块 封装性 封装 继承 私有 实例 对象
信息	信息	加密 令牌 信息论 用户名 口令 登录 信息 信道 二进制 编码 解码 局部 译码 补码 误码 纠错 空串
算法	算法	队列 索引 字节 标记 字段 算法 二分法 数据项 回溯 堆栈 节点 调度 优先级 虚拟
	执行	交换 轮转 流水号 时序 线性 执行 标识符

Fig. 2. Parent-child relationships among terms

5 Conclusion

In this paper we present an approach to automatically mining domain-dependent ontologies from corpora based on term extraction and relationship discovery technology. There are two main innovations in our approach. One is extracting terms using log-likelihood ratio, which is based on the contrastive probability of term occurrence in domain corpus and background corpus. The other is fusing together information from multiple knowledge sources as evidences for discovering particular semantic relationships among terms. In our experiment, we improve the traditional k-mediods algorithm for multi-level clustering. We have applied our approach to produce an ontology for the domain of computer science and obtained promising results.

In the future work, we will consider developing heuristic methods or rules to label the exact relations among terms and finding out a more effective ontology evaluation methodology.

Reference

1. Inderjeet Mani.Automatically Inducing Ontologies from Corpora. Proceedings of CompuTerm 2004: 3rd International Workshop on Computational Terminology, OLING'2004, Geneva.
2. A. Maedche and S. Staab. Mining ontology from text. 12th International Workshop on Knowledge Engineering and Knowledge Management,2000 (EKAW'2000).
3. A. Maedche and S. Staab. The TEXT-TO-ONTO Ontology Learning Environment. Software Demonstration at ICCS-2000-Eight International Conference on Conceptual Structures. August 14-18, 2000, Darmstadt, Germany.
4. Dr.Sadanand Srivastava, Dr.James Gil de Lamadrid. Extracting an ontology from a document using Singular Value Decomposition. ADMI 2001.
5. Lin, D. and Pantel, P. 2001. Induction of semantic classes from natural language text. In Proceedings of SIGKDD-01. pp.317-322. SanFrancisco, CA.
6. Dunning.T. Accurate Methods for the Statistics of Surprise and Coincidence, Computational Linguistics, 19(1); 61-74, March 1993.
7. Chakravarthi S Velvadapu, Document Ontology Extractor, Applied research in Computer science, Fall-2001.
8. A. Maedche and S. Staab. Discovering conceptual relations from text. In Proceedings of ECAI-2000. IOS Press, Amsterdam, 2000.
9. D. Faure and C. Nedellec. A corpus-based conceptual clustering method for verb frames and ontology acquisition. In LREC workshop on adapting lexical and corpus resources to sublanguages and applications, Granada, Spain, 1998.
10. Doan, A., Madhavan, J., Domings, P. and Halevy, A. 2002. Learning to Map between Ontologies on the Semantic Web. WWW'2002.