基于用户查询日志的命名实体挖掘*

翟海军　郭嘉丰　王小磊　许洪波

中国科学技术大学 计算机科学与技术系 安徽 230027

E-mail: zhaihaijun@software.ict.ac.cn

摘 要： 本文研究了针对大规模查询日志中丰富的命名实体的挖掘技术。已有的研究工作提出了一种基于种子实体的抽取框架，利用实体间的分布相似度来进行挖掘。然而该工作只有当种子实体仅属于单个语义类别时才能取得好的结果，而实际命名实体却往往可能从属于多个类别。本文通过引入一个弱指导话题模型，利用少量的人工指导信息，很好地解决了实体的类别模糊性，提高了挖掘的有效性。实验表明我们的方法在实体挖掘性能上显著优于已有的方法。

关键词： 命名实体，用户查询日志，话题模型

Mining Named Entities from Query Logs

Zhai Haijun，Guo Jiafeng，Wang Xiaolei，Xu Hongbo

Department of Computer Science and Technology, University of Science & Technology of China, Anhui 230027

E-mail: zhaihaijun@software.ict.ac.cn

Abstract: This paper addresses the problem of mining named entities from query logs. Previous work proposed a seed-based framework to mine named entities from query logs by leveraging distribution similarity. However, this framework works well when each named entity only belongs to one semantic class. In fact, named entities may often belong to multiple classes. In this paper, we introduce a weakly-supervised topic model to resolve class ambiguity of named entities by leveraging weak supervision from human. In this way we can greatly improve the effectiveness of the mining framework. The experiment results show that our approach significantly outperforms the previous method.

Keywords: named entity, query log, topic model.

1 引言

近年来，数据挖掘领域的一个共同的发展趋势是对大规模数据信息抽取技术的研究，尽管这些研究工作在抽取的目标信息、底层算法以及使用工具上可能各有不同。其中，用户查询日志作为一类富含大众智慧的海量数据资源，成为了数据挖掘领域广泛关注的研究对象。从查询日志中获取的各种知识不仅可以为信息检索领域所用，还可以成为机器翻译、自然语言处理等等领域的基础。本文研究了针对大规模查询日志中丰富的命名实体的挖掘技术。对查询日志中命名实体的挖掘在垂直搜索，查询推荐，以及 Web 检索等方面都有广泛的应用前景。例如用户提交了查询“the family stone”，基于命名实体挖掘的结果可以知道“the family stone”是指一部电影，通过将该查询提交到影视相关的垂直搜索中，返回的查询结果可以更好地满足用户的查询需求。

以往对命名实体识别的研究主要集中在文本领域中[1-3]，至今已有近二十年的发展历史。它作为自然语言处理领域的一项重要技术，已经取得了很多成果。早期命名实体识别的技术通常依赖于人工指定规则。近年来，机器学习的方法也开始被应用于命名实体识别，包括了监督学习[1]，半监督学习[2]和无监督学习[3]。

*本课题受国家重点基础研究计划（973）课题“大规模文本内容计算（2004CB318109）”和国家高技术研究发展计划（863）项目“网络文本的倾向性分析（2007AA01Z441）”资助。

与文本领域中的命名实体识别不同，用户查询通常都很简短（往往只有 2-3 个词），并且不具备严格的语法，语义很模糊，因此文本领域中的命名实体识别技术不能直接有效地应用到查询上。这给基于用户查询的命名实体挖掘的研究工作提出了新的挑战。已有的研究表明用户查询数据具有一些独有的分布特性，分析这些特性有助于我们从用户查询日志中挖掘命名实体。Pasca[4]提出了一种利用查询模板从用户查询日志中挖掘命名实体的确定性方法（Determ）。作者将查询分解为两部分，某个类别的实例（即命名实体）和查询模板（即查询上下文）。在此基础上，通过人工给定目标类别下的一组种子（命名实体）作为指导，估计目标类别下的查询模板的分布，来从用户查询日志中挖掘目标类别下的新命名实体。然而该工作的一个主要的局限性在于他们的方法是确定性的，只有当种子实体仅属于单个语义类别时才能取得好的结果。而实际中，命名实体往往可能从属于多个语义类别。比如，命名实体“harry potter”，在用户查询“harry potter poster”中指得是“harry potter”这部电影；而在用户查询“harry potter author”中指得是“harry potter”这本书。这样的类别模糊性使得Determ方法对实体抽取的性能受到了很大影响。

因此，本文针对命名实体的类别模糊性，提出了一个基于弱指导话题模型的命名实体挖掘框架。和Determ方法相似，该框架也利用实体的分布相似性，基于种子实体来从大规模查询日志数据中挖掘多个语义类别的命名实体。不同的是我们的方法通过引入一个弱指导的生成概率模型来学习各个类别的查询模板分布，很好地解决了命名实体的类别模糊性。这里所使用的话题模型，是关联话题模型CTM[5]（Correlated Topic Model）的一个衍生，称为弱指导关联话题模型WSCTM（Weak-Supervised Correlated Topic Model）。但是，不同于使用无指导学习的CTM模型，WSCTM可以利用少量人工标注的命名实体的弱指导话题信息来指导学习，从而确保结果话题与任务所关注的话题一一对齐。通过对该话题模型的使用，我们可以更加准确地估计目标类别下查询模板的分布，从而更好地挖掘命名实体。

在文章最后的实验中，我们收集了来自一个商业搜索引擎的 1500 万条真实用户查询数据，通过人工标注了少量种子命名实体，来训练弱指导话题模型，并将我们的方法与已有的方法相比较，实验结果表明我们的方法在实体挖掘性能上显著优于已有的方法。

2 命名实体挖掘框架

这一节介绍基于弱指导话题模型的命名实体挖掘框架。给定一组目标类别和类别下一组作为种子的命名实体，命名实体挖掘的目标是从用户查询日志中大规模挖掘目标类别相关的新命名实体，无需任何其它领域相关知识。我们的命名实体挖掘框架可以分为三个阶段：

（1）首先选择一组命名实体作为种子，并且对每个种子指派类别。考虑到命名实体往往从属于多个类别，因此这里的种子命名实体可能同时被指派一个或多个类别，比如命名实体“American pie”同时标注了“Movie”和“Music”两个类别。这里，我们对命名实体进行标注，而不是针对单个的用户查询，并且只需标注少量的种子命名实体，确保了标注工作可以简单进行。

（2）针对每个种子命名实体获取相应描述文档，然后采用弱指导话题模型来学习话题模型。具体来说，对每个种子命名实体，我们通过遍历整个查询日志，收集所有包含该命名实体的用户查询。类似于Determ方法，这里将查询分解为两部分，命名实体和查询模板。例如命名实体“harry portter”，对于查询“harry portter poster”，相应的查询模板是“# poster”。通过组合该命名实体的所有查询模板，得到该命名实体的描述文档。我们可以像普通文档一样来看待描述文档，区别在于这里查询模板被当成文档中的“词”。例如我们在查询日志中找到两个包含命名实体“harry portter”的查询，分别是“harry potter review”和“harry potter author”，则相应的描述文档包含两个“词”，分别是“# review”和“# author”。这样，对所有种子命名实体，通过遍历整个查询

日志，匹配得到相应的描述文档集合，作为我们的训练集。对训练集中的每个描述文档，将其相对应的命名实体的标注信息作为该描述文档的标注信息，然后采用我们提出的弱指导话题模型来学习，这样我们可以得到每个目标类别中查询模版的概率分布的估计。

（3）获取候选命名实体，并采用分布相似度计算来排序候选命名实体。将第二阶段中所获取的所有种子命名实体的查询模板作为目标串，遍历整个查询日志通过匹配获取候选命名实体及相应的描述文档。结合前面学习得到的每个目标类别中查询模版的概率分布的估计，通过计算目标类别和候选命名实体的描述文档间的相似度排序候选命名实体，来挖掘目标类别下的新命名实体。

上面我们详细描述了基于弱指导话题模型的命名实体挖掘框架，该框架中通过弱指导的话题模型来学习各个类别的查询模板分布，很好地解决了命名实体的类别模糊性，同时采用分布相似度计算来挖掘目标类别的新命名实体。整个框架简洁有效，其中关键部分是弱指导话题模型。

3 话题模型

这一节，我们将详细地介绍弱指导话题模型，该话题模型是已有关联话题模型（CTM）的扩展。我们在前面已提到，在命名实体挖掘任务中，类别（即话题）是预先给定，例如“Book”、“Music”等等。如何让话题模型中学习得到的结果话题（即类别）与任务所关注的话题一一对齐是首要解决的问题。然而，已有的CTM是一个隐式话题模型，不能确保结果话题和预先给定话题之间对齐。因此，我们提出了弱指导关联话题模型（WSCTM），通过利用种子命名实体的标注信息指导学习，有效地解决话题对齐的问题。

3.1 关联话题模型

我们首先简要回顾已有的关联话题模型。话题模型是一种生成概率模型[5-7]，该模型假定每篇文档中的词都是由一组话题混合生成，每个话题都是词空间上的一个分布。CTM是在早期话题模型LDA[7]的基础上提出来。相对于LDA模型，CTM可以获取不同话题之间的关联关系，更好地拟合文档集合。

给定由 M 个文档组成的集合 $\mathcal{D}=\{\omega_1,...,\omega_M\}$，这组文档共享 K 个话题，并且每个文档可以用包含 N 个词的序列来表示 $\omega=\{\omega_1,...,\omega_N\}$。在CTM中，文档集合 $\mathcal{D}$ 中每篇文档 ω 的生成过程如下：

(1) Draw $\eta \sim N_k(\mu,\Sigma)$

(2) For each of the N words ω_n

(a) Draw topic assignment $z_n \sim Multinomial(f(\eta))$，$p(z\mid\eta)=\exp\{\eta^T z-\log(\sum_{i=1}^{k} e^{\eta_i})\}$.

(b) Draw word $\omega_n \sim Multinomial(\beta_{z_n})$，$\beta$ is a $k\times V$ matrix.

整个过程的概率图模型如图1所示。给定参数 μ，Σ 和 β，文档的生成概率如下：

$$p(\omega\mid\mu,\Sigma,\beta)=\int p(\eta\mid\mu,\Sigma)(\prod_{n=1}^{N}\sum_{z_n} p(z_n\mid\eta)p(\omega_n\mid z_n,\beta))d\eta \tag{1}$$

最后，将所有文档的生成概率相乘，我们可以得到文档集合的生成概率如下：

$$p(\mathcal{D}\mid\mu,\Sigma,\beta)=\prod_{d=1}^{M}\int p(\eta_d\mid\mu,\Sigma)(\prod_{n=1}^{N_d}\sum_{z_{dn}} p(z_{dn}\mid\eta)p(\omega_{dn}\mid z_{dn},\beta))d\eta_d \tag{2}$$

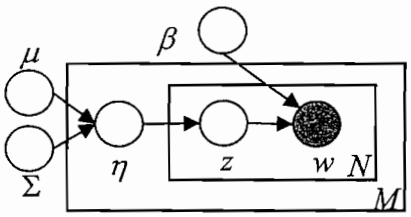

图 1 CTM 的图模型表示

3.2 弱指导关联话题模型

在我们的命名实体挖掘任务中，每个种子命名实体的描述文档都包含一个或多个类别标签。这些信息可以很好地指导算法的学习过程。WSCTM采用弱指导的方式，将已有的标注信息作为修正项，添加到目标函数来对齐结果话题和任务所关注的话题。

这里我们采用向量$\boldsymbol{y}=\{y_1,...,y_K\}$来表示每篇文档的标注信息，其中$K$表示话题数，当文档被指派了第$i$话题时，$y_i$的取值为1，否则为0。我们通过修正已有的CTM，提出一个基于弱指导的关联话题模型（WSCTM），来建模给定了类别标签的文档集合。新的文档生成目标函数定义如下：

$$\mathcal{O}(\boldsymbol{w},\boldsymbol{y})=\mathcal{L}(\boldsymbol{w})+\rho\mathcal{R}(\boldsymbol{y}) \tag{3}$$

其中ρ是修正参数，$\mathcal{L}(\boldsymbol{w})$是文档的对数似然$\mathcal{L}(\boldsymbol{w})=\log p(\boldsymbol{w}\mid\mu,\Sigma,\beta)$和定义在文档标签上的修正项$\mathcal{R}(\boldsymbol{y})$

$$\mathcal{R}(\boldsymbol{y})=\sum_{i=1}^{K}y_i p(c_i\mid\boldsymbol{w}) \tag{4}$$

其中$p(c_i\mid\boldsymbol{w})$表示模型学习文档$\boldsymbol{w}$的第i话题的概率$p(c_i\mid\boldsymbol{w})=\frac{1}{N}\sum_{n=1}^{N}z_n^i$，其中$z_n^i$表示第$n$个词是否被指派了第$i$个话题，是则$z_n^i$的取值为1，否则为0。将所有文档生成目标函数相加，可以得到整个文档集的总目标函数如下：

$$\begin{aligned}\mathcal{O}(\mathcal{D},\mathcal{Y})&=\mathcal{L}(\mathcal{D})+\rho\mathcal{R}(\mathcal{Y})\\&=\sum_{d=1}^{M}\log p(\boldsymbol{w}_d\mid\mu,\Sigma,\beta)+\rho\sum_{d=1}^{M}\sum_{i=1}^{K}y_{di}p(c_{di}\mid\boldsymbol{w}_d)\end{aligned} \tag{5}$$

在公式（5）中，我们通过修正已有的CTM，给出了WSCTM的目标函数。在该目标函数中$\mathcal{L}(\mathcal{D})$衡量话题模型生成数据的概率，$\mathcal{R}(\mathcal{Y})$约束每篇文档的话题分布集中在相应的标注话题上。修正参数ρ指示文档遵循文档标注的程度。当ρ等于0，WSCTM退化为CTM。

3.3 参数推导和估计

这一节我们详细地介绍 WSCTM 的参数估计。WSCTM 通过最大化目标函数进行参数估计。给定文档$\boldsymbol{w}$和模型$\{\mu,\Sigma,\beta_{1:K}\}$，每个文档隐变量的后验分布是

$$p(\eta,z\mid\boldsymbol{w},\mu,\Sigma,\beta_{1:K})=\frac{p(\eta\mid\mu,\Sigma)\prod_{n=1}^{N}p(z_n\mid\eta)p(w_n\mid z_n,\beta_{1:K})}{\int p(\eta\mid\mu,\Sigma)(\prod_{n=1}^{N}\sum_{z_n}p(z_n\mid\eta)p(w_n\mid z_n,\beta))d\eta} \tag{6}$$

如同 CTM，该公式难于计算。因此，我们采用变分期望最大化（Variational Expectation-Maximization）方法来估计参数。

我们采用 Jensen 不等式来计算文档生成目标函数的下界：

$$\log p(w \mid \mu, \Sigma, \beta) + \rho \mathcal{R}(y) \geq \mathrm{E}_q[\log p(\eta \mid \mu, \Sigma)] + \sum_{n=1}^{N} \mathrm{E}_q[\log p(z_n \mid \eta)] \tag{7}$$
$$+ \sum_{n=1}^{N} \mathrm{E}_q[\log p(w_n \mid z_n, \beta)] + \mathrm{H}(q) + \mathrm{E}_q[\rho \mathcal{R}(y)]$$

其中的期望基于隐变量的变分分布q来求取，$\mathrm{H}(q)$表示分布的熵。

$$q(\eta_{1:K}, z_{1:K} \mid \lambda_{1:K}, \nu^2_{1:K}, \phi_{1:N}) = \prod_{i=1}^{K} q(\eta_i \mid \lambda_i, \nu_i^2) \prod_{n=1}^{N} q(z_n \mid \phi_n) \tag{8}$$

最后，在 E（期望）步中，我们最大化公式(7)求变分参数ζ，λ，ν_i^2和ϕ。这里，我们采用梯度下降算法，通过迭代最大化目标函数求取各变分参数。各变分参数的更新函数如下：

$$\hat{\phi}_{n,i} \propto \exp\{\lambda_i + \frac{\rho}{N} y_i\} \beta_{i,w_n} \qquad \hat{\zeta} = \sum_{i=1}^{K} \exp(\lambda_i + \nu_i^2 / 2) \quad .$$

$$d\mathcal{L}/d\lambda = -\Sigma^{-1}(\lambda - \mu) + \sum_{n=1}^{N} \phi_{n1:K} - (N/\zeta) \exp(\lambda + \nu^2/2)$$

$$d\mathcal{L}/d\nu_i^2 = -\Sigma_{ii}^{-1}/2 - N/2\zeta \exp(\lambda_i + \nu_i^2/2) + 1/2\nu_i^2$$

在 M（最大化）步中，我们最大化公式(5)来求取模型参数μ,Σ和β。所有参数的更新公式如下：

$$\hat{\beta}_i \propto \sum_d \phi_{d,i} n_d \qquad \mu = \frac{1}{\mathcal{D}} \sum_d \lambda_d \qquad \Sigma = \frac{1}{\mathcal{D}} \sum_d I\nu_d^2 + (\lambda_d - \hat{\mu})(\lambda_d - \hat{\mu})^T$$

上面我们详细地介绍了 WSCTM 的学习过程。我们可以将训练好的 WSCTM 用于预测新文档的话题分布。预测的程序与 CTM 的预测程序相同，限于篇幅这里不再赘叙。

4 实验结果及其分析

我们收集了来自一个商用搜索引擎的真实用户查询数据，通过实验来验证我们方法的有效性。实验中，我们与已有的Determ方法做了比较。Determ方法的基本思想，假定每个命名实体只属于一个类别，通过统计每个类别下所有种子命名实体的查询模板的分布，作为该类别的查询模板的真实分布的估计，通过计算候选命名实体的描述文档与类别查询模型的估计分布之间的相似度来排序候选命名实体。在这里，WSCTM的修正参数ρ设置为1。

4.1 实验数据

本文所采用的实验数据集包含了1500万条真实用户查询数据，这些查询数据是从一个商用搜索引擎随机采样得到。该数据集包含6,623,961个不同的用户查询，用户查询的平均长度为2.423个英文单词。这里，用户查询被认为是互相独立的，不考虑它们是否来自同一用户。

本实验中，我们共考虑了10个不同语义类别。这样的实验设置，综合考虑了不同粒度的类别，其中包括命名实体识别和挖掘中经常会涉及的粒度较粗的类别，比如“Location”，“Person”和“Food”等，也引入了一些粒度相对较细的类别，比如来自“Entertainment”的“Movie”，“Music”和“Game”。对于这些类别，我们从不同的网站收集了150个种子命名实体，这些网站包括有Amazon（www.amazon.com），GameSpot（www.gamespot.com）和Lyrics（www.lyrics.com）等等。我们召集了3个研究生来标注这些命名实体，对标注不一致的命名实体，采用投票的方式来确定命名实体最终的标签（多于2个人同时标注了的标签被接受）。其中有31个种子命名实体同时被标注了多个类别（占据总数的20.7%），比如“lord of the rings”同时属于“Movie”和“Game”。

每个类别中都有标注多类别的命名实体，多类别命名实体所占比例随类别而不同（≥12.5%），其中类别“Book”中多类别命名实体所占比率最大（60.6%），类别“Software”中多类别命名实体所占比例最少（12.5%）。此外，类别间的命名实体的重叠率（重名命名实体所占比

率）各不相同，其中“Movie”和“Book”，“Movie”和“Game”两对类别的重叠率最高（≥24%）。这符合实际中，电影经常由同名书籍改编而来，而后又改编制作出同名游戏的现象。比如电影“Harry Potter”就是由同名书籍改编拍摄的，而后又通过改编制作出了同名游戏。

4.2 评价方法

为了确保评价的公平性，这里将我们的方法得到结果和Determ方法的结果，按类别各取前250个相混合，然后通过人工评判各类别下的结果命名实体（对评价人员隐藏了结果的来源），对属于该类别的结果命名实体判定为true，否则判定为false。另外，为了确保结果的准确性，我们同时召集了3个研究生来做评价，对评价结果不一致的命名实体，采用投票的方式来确定命名实体最终的评价结果（多于2个人同时认同的结果将被接受）。基于上述人工评价后的结果数据，我们采用前N个结果的准确率P@N来度量在每个类别下各算法的性能，P@N表示在结果列表的前N个结果中人工判断为true的实例所占比率。

4.3 实验结果及其分析

采用上面所描述的实验数据和度量，将我们的方法（Our）与Determ方法相比较。实验结果如表1所示。从表1可以看出，在所有类别下，我们方法的准确率P@N都优于（或等于）Determ方法，并且我们方法的P@250平均准确率（0.912）相对Determ方法的P@250平均准确率（0.799）显著提高了14.1%（统计显著性$P<0.01$）。此外，Determ方法的准确率P@N随结果数N的增加急剧下降，而我们的方法所产生的结果相对要稳定很多。需要特别指出的是，对于类别“Location”和“Company”，两个方法所得到的结果都非常好（我们方法的结果略好），这可能是因为类别“Location”和“Company”下命名实体对应的查询上下文非常丰富，而且这些查询上下文对类别的指示作用也很明确，例如类别“Location”下的查询模板“# daily news”、“map of #”等等。

这里我们进一步分析了我们的方法能够取得更好的挖掘效果的原因，这主要是由于我们的方法通过考虑命名实体的多类别特性，更加准确地估计出各目标类别的查询模板分布。表2中给出了我们的方法和Determ方法所估计的各类别下前5个查询模板（根据各类别下查询模板出现的概率从大到小取前5个，限于篇幅这里只取了前5个）。从表2可以很直观地看出，Determ方法中，各类别下的前5个查询模板中往往混杂有其它类别的查询模板，比如类别“Book”的第2个模板“# movie”。这是因为Determ方法忽略了命名实体的类别歧义性，将同时属于类别“Book”与“Movie”的命名实体对应的查询上下文都强行归于“Book”类别下；与之相反，在我们的方法中，由于考虑命名实体的多类别特性，使得各类别下的查询模板分布学习地更加准确。

表1 各类别下准确率P@N对比

目标类别	P@25		P@50		P@100		P@150		P@250	
	Our	Determ	Our	Determ	Our	Determ	Our	Determ	Our	Determ
Movie	1.00	1.00	1.00	0.98	1.00	0.90	0.98	0.87	0.95	0.79
Book	1.00	1.00	1.00	1.00	0.99	0.94	0.97	0.90	0.96	0.82
Game	1.00	1.00	1.00	1.00	0.98	0.96	0.96	0.93	0.94	0.85
Music	0.96	0.92	0.94	0.90	0.93	0.82	0.91	0.75	0.90	0.69
Software	0.96	0.96	0.96	0.90	0.93	0.90	0.91	0.85	0.87	0.79
Location	1.00	1.00	1.00	1.00	0.99	0.99	0.98	0.97	0.98	0.96
Food	0.96	0.88	0.92	0.86	0.90	0.80	0.88	0.80	0.86	0.78
Company	1.00	0.96	1.00	0.96	0.98	0.96	0.98	0.96	0.98	0.95
Animal	0.96	0.84	0.92	0.76	0.86	0.69	0.84	0.64	0.78	0.58
Person	0.96	0.96	0.96	0.96	0.96	0.96	0.92	0.87	0.90	0.78

表2 各类别下前5个查询模板对比

Movie		Book		Game		Music		Software	
Our	Determ	Our	Determ	Our	Determ	Our	Determ	Our	Determ
# movie	# movie	# book	# book	# cheats	# cheats	# lyrics	# lyrics	# download	# download
# the movie	# the movie	# summary	# movie	# walkthrough	# movie	lyrics to #	lyrics to #	download #	download #
# trailer	# trailer	# books	# summary	# cheat codes	# walkthrough	# soundtrack	lyrics #	free #	free #
# soundtrack	# soundtrack	# quotes	# quotes	# game	# games	# song lyrics	#soundtrack	# downloads	# downloads
movie #	# pictures	# review	# books	# games	# toys	lyrics #	# quotes	# updates	windows #
Location		Food		Company		Animal		Person	
Our	Determ	Our	Determ	Our	Determ	Our	Determ	Our	Determ
# daily news	# movie	# recipes	# recipes	# computers	# computers	# pictures	# pictures	# movies	# movies
# news	map of #	# salad	# salad	# support	# support	# list	# list	# biography	# biography
# lottery	# daily news	# diet	# diet	# laptops	# laptops	# breeds	# breeds	# quotes	# quotes
city of #	# news	grilled #	# grilled	# parts	# parts	# sex	# recipes	# pictures	# pictures
map of #	# lottery	# salad	# recipe	# financial	# dealers	# names	# shoes	pictures of #	pictures of #

5 结论

查询日志是大量用户长期查询行为的记录，其中包含丰富的知识。本文研究了对大规模查询日志中命名实体的挖掘技术。与文本领域的命名实体挖掘不同，用户查询通常很简短，并且含有大量的噪音。因此文本领域中的命名实体识别技术不能直接有效地应用到查询上。这给基于用户查询的命名实体挖掘的研究提出了极大的挑战。但是，用户查询数据具有一些独有的分布特性。并且实际中，命名实体往往可能从属于多个类别。本文提出了一个弱指导话题模型的命名实体挖掘框架，该框架下通过一个弱指导的生成概率模型来学习各个类别的查询模板分布，很好地解决了命名实体的类别模糊性，同时结合实体的分布相似性，来提高挖掘的有效性。文章最后通过实验验证了我们方法的优越性。下一步我们期望探索挖掘得到的大量命名实体在检索方面的应用。

参 考 文 献

[1] Borthwick Andrew, Sterling J., Agichtein E., Grishman R. NYU: Description of the MENE Named Entity System as used in MUC-7. In Proc. Seventh Message Understanding Conference. 1998.

[2] Cucchiarelli Alessandro, Velardi P. Unsupervised Named Entity Recognition Using Syntactic and Semantic Contextual Evidence. Computational Linguistics 27:1.123-131, Cambridge: MIT Press. 2001.

[3] Evans Richard. A Framework for Named Entity Recognition in the Open Domain. In Proc. Recent Advances in Natural Language Processing. 2003

[4] Paşca, M. 2007. Weakly-supervised discovery of named entities using web search queries. In Proceedings of the Sixteenth ACM Conference on Conference on information and Knowledge Management, 2007.

[5] D. M. Blei and J. D. Lafferty. Correlated topic models. In In Proceedings of the 23rd International Conference on Machine Learning, pages 113-120, 2006.

[6] T. Hofmann. Probabilistic latent semantic indexing. In SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pages 50-57,1999.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3:993-1022, January 2003.