上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

# 学士学位论文

BACHELOR'S THESIS


论文题目：基于海量摘要文本的话题树构建

学生姓名: 吴昊

学生学号: 5140219226

专　　业: 信息工程

指导教师: 周燕

学院(系)： 电子信息与电气工程学院

# 基于海量摘要文本的话题树构建

## 摘要

关键词分类不仅仅是语料库分析中非常基本的任务，而且其在信息过滤、推荐系统以及搜索引擎中均具起到非常关键的作用。现有的方法是基于模式的方法，提取上下文语义将句子的组织成分进行分类。但是，通过将每个词组作为一个独立的概念节点来考虑，会更加重视主题临近性和语义相关性。在本篇文章中，我们采用了一种多层次主题构建的分类方法TTG (Topic Tree Generator)，其中每个节点代表一个概念性主题以及一组语义相关的主题聚类。该分类方法使用了词向量和球形聚类的方法通过递归的方式完成了多层次的主题构建。在递归过程中，我们采用了包括：（1）一种优化的球形聚类模型，可将主题聚类在向更小颗粒度的主题分层时确保将部分词组合理下移；（2）一种局部的文本构建方案，通过在更小颗粒度的文本下训练词向量，提高模型在向更小颗粒度主题分层时的准确性。在实验部分我们主要采用定性化描述和基于用户调研的定量化描述来完成对模型现的评价，综合考量了模型结果的关系准确性和术语一致性指标。

**关键词：**TTG，多层次主题构建，词向量，球形聚类

# TTG: TOPIC TREE GENERATOR BY ANLYZING ABUNDANT PAPER TITLE AND ABSTRACT


## ABSTRACT

Constructing a hierarchical topic tree is not only a very basic task in corpus analysis, but also plays a key role in information filtering, recommendation systems, and search engines. Existing pattern-based methods classify the organizational components of sentences by extracting contextual semantics. However, by considering each phrase as an independent concept node, the topic proximity and semantic relevance will be more emphasized. In this article, we use a TTG (Topic Tree Generator), a multi-level topic-building classification method, in which each node represents a conceptual topic and a cluster of semantically related topics. The classification method uses word vectors and hierarchical clustering to accomplish multi-level topic construction in a recursive process. In the recursive process, we have adopted: (1) an optimized spherical clustering model that can ensure that certain word combinations are shifted downwards when subject clustering is layered to smaller grained topics; (2) A local text construction scheme improves the accuracy of the model when stratified to a smaller granularity theme by training word vectors under a smaller granularity of text. And in the experiment section, we mainly utilize the qualitative description and the quantitative description based on the user study to complete the evaluation of the model, and comprehensively consider the accuracy of the relationship between the model results and the term consistency.



**Key words:** TTG, topic Tree, term embedding, spherical clustering

# 目 录

# 第一章 绪论

从非结构化文本中实现话题树的构建在语义分析领域是一个非常基础且重要的工作，并且在非常多的领域都有相应的应用。举例来说，从一系列非结构化的论文文本中实现论文研究话题的抽取、聚类、分层能引导用户快速发现其感兴趣的部分，获得有益的知识。并且在推荐系统方面，由于推荐内容包含大量的文本化描述，如果能有效实现话题树的构建，那么会根据推荐内容话题在话题树中的精准定位，极大帮助到推荐内容的排序，即根据询问文本返回更准确的推荐信息[22]。

## 1.1. 研究现状

现有对于话题的聚类分层方法主要将树状结构中每个节点表征一个独立的话题[8]。其主要采用预定义词组字典的方法从文本中抽取话题，如$A$属于$B$，$B$包含$A$的方式将抽取出来的话题根据预定义的语义关系进行聚类分层，并将每个坏头赋值给树状结构中的每个节点。其优点是具有较高的抽取精度以及分层精度，但是由于其是需要预先构建语义关系，所以其存在以下三点不足：（1）低覆盖率：由于无法覆盖所有的语义关系，只有抽取出满足预设的语义关系的话题对，所以会使得话题集合不完整。（2）高冗余度：由于一个话题可能有多种表达方式，如单数复数的差异、缩写的差异无法排除。（3）低信息量：由于每一个节点仅仅代表一个话题，无法有效表达该话题的完整信息量。

## 1.2. 相关工作

在本部分我们回顾一下现有的对于话题树构建的方法，包括三大部分，（1）基于模式的方法，（2）基于聚类的方法，（3）监督式方法。

（1）基于模式的方法：Hearst 模式法，如基于 NP such as NP 和 NP and NP 等特定模式可从文本数据中自动抽取出上下文语义关系[8]。 然后，基于更多相关词汇模式的整理和设计，该方法可从网络语料库[2,17,19]或维基百科[7,18]中提取更多的关系对。 随着词汇模式滚雪球式的发展，研究人员可以告诉机器如何使用统计方法在大量文本语料库中获取知识[1,23]。与此同时，Carlson 等人在 2010 年提出了永无止境语言学习（NELL）的学习架构[3]。Patty 利用解析法派生出具有不同类型的关系模式，并将这些模式归类到特定关系分类中[16]。 近期的 MetaPAD[9]使用上下文感知短语分割来生成高质量的模式关系，并将同义模式组合在一起以获得特定关系的大量集合。基于模式方法已经证明了它们基于手工制定的规则或生成的模式发现特定关系的有效性。 然而，即使语法结构都能被人们发现以及整合然而，用计算机来对相应的语法结构进行解析重构也是一件极为困难的事情。因为计算机编程语言和人类的自然语言语法结构完全不同，前者是人为设计语言，为了便于编程人员的任务共享和程序编写，设计之初即将编程语言设为上下文无关联的，而自然语言在进化过程中逐渐产生了词义和前后文相关联性，是比较复杂的。两者语言的理解体量和解析计算量完全无法相提并论。

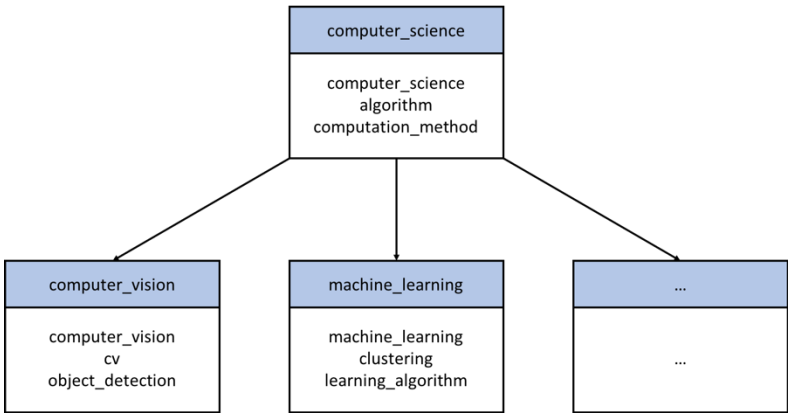
（2）基于聚类的方法：如今已经提出了大量的聚类方法来从文本数据库中构建名词分类。一般来说，聚类方法首先从大量文本中学习训练得到单词或术语的某种表现形式，如词向量形式，然后根据它们的表现相似性[2]和聚特性[8]将它们组织成特定集合聚类。Fu 等人通

过使用基于单词嵌入的单词和其上下文之间的语义关系来识别候选单词对是否具有(“is-a”)关系[6]。Luu 等人提出使用动态加权神经网络的方法通过学习词向量来识别分类关系[14]。

（3）监督式方法：对话题树的构建还有半监督学习的方法[10,11]。 大致而言，这些方法首先将单词对或术语对分类为存在特定关系和不存在特定关系这两类，并使用关键词或上下文本对的先验知识[4,6,12,21]或从 NLTK 中收集到的特定句法概念信息[13,20]实现单词对的分类。对于大量特定于某一领域的文本数据（如我们在本文中使用的会议期刊等论文数据），几乎不可能从特定专家处收集到丰富的先验知识。 因此，我们关注更关心无监督方法的发展。

## 1.3. 问题描述

本篇论文研究的问题主要是从非结构化的文本中实现对特定话题的抽取、聚类和分层工作。在此论文中我们将会扩充话题树中每个节点所表达的话题量。也就是说，在话题树中的每个节点是几个具有语义相似性的话题的唯一聚类。如图一所示，假设非结构化文本来自于计算机科学领域，那么在我们的根节点中表示的是Computer Science及其相似的话题聚类，依次往下层被分类为Machine Learning、Computer Vision等。在每个话题节点中，我们用多个语义相似的名词进行统一描述，如在Computer Vision中有computer_vision，其语义相似名词不但包含其缩写cv，还包含cv领域所涉及到的不同方面，如object_detection等。




**图一　话题树构建示例**

## 1.4. 模型说明

本文采用的TTG模型(Topic Tree Generator)实现了高精度且自动化，无需人为先验知识输入的，对任意领域文本的话题树构建过程。该模型主要将各话题放入了隐性空间计算其语义向量，并利用分层聚类的算法自顶向下递归构建话题树。由于本模型主要通过计算语义向量以及分层聚类的算法实现话题树构建，故该模型在设计过程中的挑战也来源于此。其一，在面对一个话题集合的时候如何有效且合理地进行分类，也就是说哪些话题应该被放在当前层，而哪些话题应该被分类至下一层。举例来说，在图一中，一些常用话题如cs以及computer science应该留在根节点而无需划分至下一层。因此该挑战就是要明确对于话题集合哪些应该留在当前层，哪些应该划分至下一层。其二，在通过文本进行话题向量计算时文本集合应该如何抽取，即在不同分类层次下哪些文本应该被选择计算话题向量。由于在计算话题向量过程中会通过学习选择的文本集合来获得话题向量间的相似度，因此具

有相似文本位置出现概率的话题会具有极高的语义相似性。然而假如我们始终拿全部的文本计算话题向量，那么在进行聚类分层的时候无法有效地将部分词组划分至下一层。举例来说，当对machine learning相关话题进行分层时候我们发现machine learning和reinforcement learning具有相近的话题向量，因此很难从该话题集合实现高质量的话题分类。

TTG模型主要设计了两个子模块用以解决上述两个困难。第一个是优化的球形聚类模型，实现了高质量的话题分层聚类。基于计算话题集合中各个话题与该话题集合对应的文本集合的代表性指标并对其进行排序实现，由排序结果判断该话题应该留在这一层还是被划分入下一层。第二个是基于话题集合的文本集合选取，实现了在不同层次聚类分层下话题向量的更新。为了在不同层次最大限度地找到话题向量的差异，我们不选择使用全体文本用于话题向量的训练，而选择基于该层话题集合抽取的小颗粒度文本集合进行训练。使用该文本集合的好处就d在于能以更高的精度捕捉到话题语义，并且不会受到与此话题集合无关的其他话题的干扰。因此，即使是在较低的分类等级下也能实现较好的话题划分。

## 1.5. 实验说明

本论文在实验部分使用了两份真实的数据集用于评估该模型的表现，结果显示TTG能够较好地实现较高质量的话题树构建。在定性分析方面通过图表的方式展现了模型在话题树生成方面的效果，在定量化分析方面中采取了用户调研的方法对分类结果进行数值评估。

# 第二章 问题定义

话题树构建模型的输入包含两部分：（1）全体文本$D$；（2）全体话题集合$T$。集合$T$中所有话题都是从全体文本集合$D$中利用特定工具得到的专业术语，组成了话题树构建所需要的所有话题。基于文本集合$D$和话题集合$T$，我们希望构建一个具有树状结构的，自顶向下的话题树$H$。对于树$H$中的每个节点$C$均代表一些具有相似语义的话题的聚类，用$T_C$表示。假设$C$的子节点我们定义为$S_C = \{S_1, S_2 \dots S_n\}$，那么每个$S_i$即表示$C$的子话题集合。

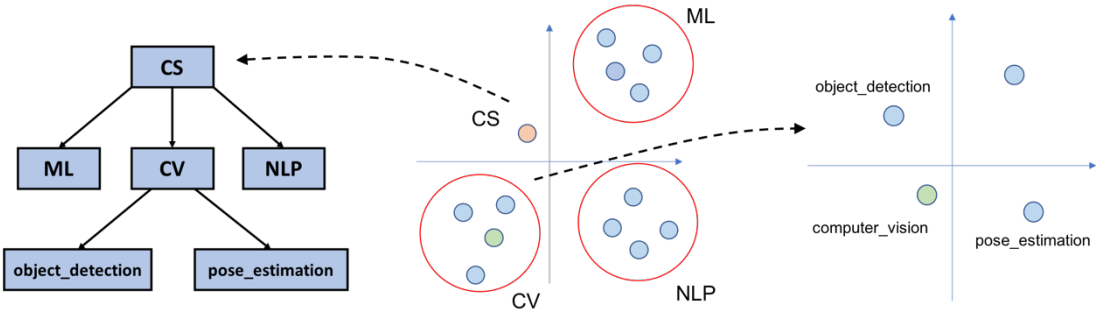
# 第三章 模型说明

在本章节，我们会着重描述下该论文中使用的TTG模型的实现方式。本章节共可分为4个部分，3.1介绍模型概览、3.2介绍话题抽取的依据算法和实现方法、3.3介绍话题聚类分层算法，即优化的球形聚类模型、3.4介绍文本选取的依据和实现，即如何选择基于某层话题集合抽取相应文本集合四个方面。

## 3.1. 模型概览

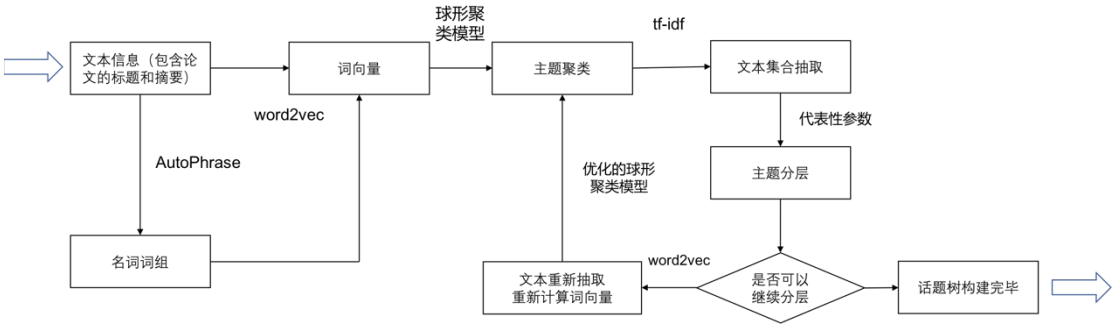
该模型主要将各话题放入了隐性空间计算各自话题的语义向量，并利用分层聚类的算法自顶向下递归构建话题树。如图二所示，在顶层节点我们首先初始化假设所有话题$T$都属于该节点$C$，且这些话题代表了全体文本$D$。此时，我们即开始采用分层聚类的算法自顶

而下递归构建话题树，递归过程中根据层级和聚类标签不断更新文本集合和话题集合，一旦话题树的层级到达我们预设的层级$L_{max}$，则递归运算结束。一旦递归运算结束，我们就可以得到一个完整的话题树结构，树结构的上下层之间的话题集合具有较高的包含关系且同一层的话题集合间存在一定的差异性。递归运算的示意图由图二给出，在不同层的话题集合下均采用优化的球形聚类模型，对话题集合进行合理的聚类，并依据代表性指标将常见话题和子话题集合区分开来，确保每一层树节点的合理构建。




**图二　TTG模型概览**

基于给定的话题集合$C$，我们会采用优化的球形聚类模型将$C$划分为多个子话题集合$S_C = \{S_1, S_2 \dots S_n\}$。在模型说明部分中我们有提到过这一步骤存在两点挑战：（1）如何有效将话题集合进行划分，即哪些话题应当留在当前层，哪些话题应该下移一层。由于一些在当前层非常普遍的话题无法代表任何一个子话题集合，应当将这些话题留在当前层而非划分至下一层，所以找出这些常见话题就是我们要解决的一个困难。（2）当我们在不同层级进行词向量计算的时候，考虑到假如不更新文本集合，那么词向量无法更新，也就是说在更小颗粒度的文本集合下进行话题聚类分层会显得极为困难。因此基于以上两点，我们必须设计合理的更新机制来完成词向量的更新。在接下来的几个模块我们会依次介绍本论文为这两个问题设计的相应的解决方案。图三表示的是TTG模型的总体实现流程。




**图三　TTG模型总体实现流程及方法概览**

## 3.2. 高质量话题抽取

本模型中，我们采用AutoPhrase[24]进行文本集合的话题抽取。AutoPhrase是一种不需要

人为建立知识数据库的话题抽取技术，即不需要专业人士预先对词组贴标签。该模型基于Robust Positive-only Distant Training和POS-Guided Phrase Segmentation技术实现话题抽取：

（1） Robust Positive-only Distant Training：

事实上，相比于人工标签标识，许多高质量的短语可以在现有知识库中免费获得，并且可以很容易产生比人类专家更大规模的知识数据库。即使对于特定领域的语料库通常也是如此。因此，对于词组抽取工作，我们利用现有的从维基百科和Freebase等知识库中获取到的高质量短语，以摆脱额外的人工标签成本。我们从给定领域语料库的知识库中独立构建正面和负面标签样本，并训练一些基础分类器。然后，我们汇总这些分类器以减少负面标签的噪音，从而获得更大样本的正面短语，从而提高分类器的分类质量和分类速度。

（2） POS-Guided Phrase Segmentation：

在特定领域方面，如果没有特定领域的语言知识，短语的抽取准确性可能会受到一定的限制。但是如果方法不考虑语言在特定领域的特殊性，则很难支持多种领域环境的短语抽取任务。在精确度方面，特定的分类器可能会损害该任务的领域特性。作为折衷的办法，AutoPhrase将预先训练好的词性（POS）标记器结合到文档集合中以提高性能。POS引导的短语分词能够利用POS标签中的句法信息来帮助分词模型更准确地定位短语边界。

## 3.3. 优化的球形聚类模型

在本模型中，我们会利用优化的球形聚类模型将任意一个层次下的话题集合进行分层聚类。该模型基于球形聚类模型[5]进行优化改进，原模型也被称为spherical k-means算法，通过计算向量间的余弦距离，最小化聚类结果下的方差总和，从而尽可能确保在同一聚类下的词组具有相同或者相似的向量方向。选择使用该模型而非普遍使用的k-means模型的原因是来源于余弦距离在计算词向量间相似度的研究中具有比较好的实际表现[15]。

3.3.1. 优化聚类过程

在普遍使用球形聚类的思想背景下，假如我们需要将一个完整的话题集合$C$进行聚类分层，常用思维下很容易就能想到直接对话题集合C中所有话题的词向量进行球形聚类，确保能将$C$中的话题向量聚类到$K$个子话题中，那么就很容易地将话题集合分类为$K$个子话题集合。该方法主要采用计算余弦相似度在向量空间内对高维向量的方法进行聚类：

$$\mathrm{d}(x,p) = 1 - \cos(x,p) = 1 - \frac{< x,p >}{||x|| * ||p||} \tag{1}$$

该方法基于词向量间的夹角度数表示了词与词之间的相似度。

事实上，使用返回文本和查询文本向量之间的角度以来一直是用于计算关键词搜索中文档的相关性排名的信息检索的首选方法。但是，这个直接使用球形聚类的策略却存在一定的问题，即并没有考虑到并不是所有子话题集合中的话题都能够代表该子话题，也就是说存在我们之前提到过的常见话题，并无法单独代表任何一个子话题，应该归属于当前层。而且，由于这些常见话题的存在，会使得在更小精度的子话题分层聚类存在更大的挑战。因为这些常见话题会较高频率出现在文本中的任意位置，其向量位置会存在于子话题聚类的边界附近，因此对于子话题在更小精度的划分会变得十分模糊，使得更小精度的子话题分层聚类存在较大的挑战与困难。当分层任务逐渐往更下层展开时，这个挑战会越来越大，也就是说根本没有可能将聚类工作完成好。

基于以上提到的困难，我们决定使用优化的球形聚类模型。如图二所示，为了解决上

述问题核心点在于设计一个合理的机制，使得该机制能够在每一次话题集合的分层聚类过程中找出当前层的常见话题，从而将其置于当前层，并划分出子话题集合，并将子话题集合置于下一层。找出常见话题和划分子话题两者是互益的，也就是说如果找出常见话题并且排除在话题集合外，那么就有利于子话题的清晰聚类；而子话题的划分能帮助检测出无法代表子话题的常见话题。那在每一层的分层过程中，假如我们都能实现对于常见话题和子话题集合的切分，对于常见话题，我们将其置于当前层，对于子话题集合我们置于下一层，并在此采用优化的球形聚类模型，再划分出常见话题和子话题集合，循环往复之下话题树就可以自然而然地得到构建，并且能够保证该话题树的构建结果具有较高的可信度。

因此下面提到的算法一则是为了解决上述问题，也就是说从一个话题集合中将常见话题和子话题集合进行清晰地划分，并通过划分完成的子话题集合得到子文本集合，更新话题向量，重新采用球形聚类完成聚类特征的抽取，如此可构建具有较高可信度的话题树。

**算法一　优化球形聚类实现步骤**

**Algorithm 1:** Adaptive clustering for topic splitting.

**Input**: A parent topic $C$; the number of sub-topics $K$; the term representativeness threshold $\delta$.

**Output**: $K$ sub-topics of $C$.

$C_{sub} \leftarrow C$;
**while** *True* **do**
  $S_1, S_2, \ldots, S_K \leftarrow$ Spherical-Kmeans$(C_{sub}, K)$;
  **for** $k$ *from 1 to* $K$ **do**
    **for** $t \in S_k$ **do**
      $r(t, S_k) \leftarrow$ representativeness of term $t$ for $S_k$;
      **if** $r(t, S_k) < \delta$ **then**
        $S_k \leftarrow S_k - \{t\}$;
  $C'_{sub} \leftarrow S_1 \cup S_2 \cup \ldots \cup S_K$;
  **if** $C'_{sub} = C_{sub}$ **then**
    Break;
  $C_{sub} \leftarrow C'_{sub}$;
Return $S_1, S_2, \ldots, S_K$;

算法一说明了优化球形聚类的实现步骤，基于话题集合$C$，我们目标是将$C$分解为$K$个聚类，并将常见话题置于当前层。首先我们将话题集合$C$赋值于$C_{sub}$，然后在每次迭代过程，先利用球形模型对$C_{sub}$分至$K$个聚类，然后对于每个聚类中的子话题集合进行遍历，计算每个子话题与该子话题集合的代表性参数$r$，假如无法代表该子话题，即$r$小于我们预先设定的阈值$\delta$，则认为该话题为常见话题，置于当前层，反之置于子话题集合中。当所有常见话题都被置于当前层，也就是说所有子话题集合中都无法再找到常见话题则退出迭代过程。最终我们就能得到该层的常见话题和$K$个子话题集合$C_{sub}$。

3.3.2.　话题代表性参数设定

在算法一的实现流程中，我们引入了话题代表性参数$r$用于表示某个话题与其所处的话题集合的代表性关系，即为了解决某个话题是否可以代表该话题集合这一问题。首先，我们设想直接利用话题的词向量与话题集合的中心向量的余弦距离作为代表性判断指标，但我们很快发现这个方法具有很大的争议性和不可靠性：常见话题的向量也可能和中心向量具有非常小的余弦距离，导致常见话题的检测出现极高的不准确性，因此我们放弃采用直接通过余弦距离由近及远的排序直接定义相关性的由高到低的方法进行相关性评定。

与之相对，我们采用了代表性参数这一个指标用于评定话题和话题集合的代表性关系。通过对该问题的理解，我们发现一个话题集合中具有较高代表性的词具有以下两点特征：（1）在对应该话题集合的文本集合中大量高频率出现；（2）在该话题集合的兄弟话题集合中出现频率较低或者不出现。因此为了能够综合考虑到这两点特征，即高频出现在对应话题文本，低频出现在兄弟话题文本中，我们必须选取出对应某一话题集合的文本集合。在这一个步骤中我们采用了TF-IDF算法获取对应话题集合$S_k$的文本集合$D_k$，在每个$S_k$集合中我们设计了两个参数来计算出话题$t$和话题集合$S_k$的代表性参数：

高频性pop：话题集合中具有较高代表性的话题应该在对应的文本集合中高频出现。

集中性con：话题集合中具有较高代表性的话题相比于该文本集合的兄弟文本集合应该与该文本集合更加相关，也就是说在该文本集合中出现的频率理应更高。

为了综合考虑到这两个参数，我们发现两者应当在计算代表性参数r的时候彼此满足，也就是说具有代表性的话题应当既满足较高出现频率，也要满足较高的集中性。因此我们定义用于表示某个话题$t$与其所处的话题集合$S_k$的代表性参数$r$满足下式：

$$\mathrm{R}(\mathrm{t}, S_k) = \sqrt{pop(t, S_k) * con(t, S_k)} \tag{2}$$

其中$\mathrm{pop}(\mathrm{t}, S_k)$和$\mathrm{con}(\mathrm{t}, S_k)$分别表示话题$t$在话题集合$\mathrm{S_k}$中的高频性和集中性。接下来我们用文本集合$D_k$表示其属于话题集合$\mathrm{S_k}$。我们设定高频性$pop$用下式表示：

$$\mathrm{pop}(\mathrm{t}, S_k) = \frac{\log(tf(t, D_k) + 1)}{\log(tf(D_k) + 1)} \tag{3}$$

其中$tf(t, D_k)$表示词组$t$在文本集合$D_k$中出现的频率，而$tf(D_k)$表示文本集合$D_k$的总字数。在对数函数中后加1是为了避免当$tf(\cdot)$取0时出现计算错误。

为了计算某个话题$t$与其所处的话题集合$S_k$的集中性，我们因为考虑到该话题应该在该话题集合的兄弟话题集合中出现频率较低或者不出现，所以我们引入了BM25相关度这一指标用于计算集中性指标con，公式表示如下：

$$con(t, S_k) = \frac{e^{rel(t, D_k)}}{1 + \sum_{j=1}^{K} e^{rel(t, D_j)}} \tag{4}$$

其中$D_k$表示属于话题集合$\mathrm{S_k}$的文本集合，$D_j$表示属于话题集合$\mathrm{S_k}$和其兄弟话题集合$\mathrm{S_k}$–组成的共$K$个子话题集合中第$j$个话题集合的对应文本集合。$rel(t,\ D_k)$是话题$t$和文本集合$D_k$间的BM25近似值。BM25算法通常会用来做搜索相关性评分，即对一个指定的$Query$返回搜索结果Document的排序，其思路主要是对Query进行语速解析，生成语素$q_i$，然后对于每个搜索结果$d$，计算每个语素$q_i$与$d$的相关性得分，最后，将$q_i$相对于$d$的相关性得分进行加权求和，从而得到Query与$d$的相关性得分，公式如下：

$$\mathrm{BM25}(\mathrm{Q}, \mathrm{d}) = \sum_{i=1}^{n} W_i * R(q_i, d) \tag{5}$$

该指标考虑到的一点就是一个词如果在一个文本集合中出现频率不高，但是在某个文本中出现频率较高，即表示这个词和这个文本关系相对比较密切，举例来说，假如文本是‘中国的蜜蜂种类繁多’，因为中国这个词在文本集合中出现频率较高，而蜜蜂这个词在文本集合中出现频率较高，则相比之下，蜜蜂这个词更能表征这个文本的内容。其次假如文本的总字数越少，则在同等情况下该词和该文本的关系应该更加密切。在BM25算法中，$W_i$表示$Query$中的语素$q_i$的权重，$R(q_i, d)$表示语素$q_i$与文档$d$的相关性得分。

$$W_i = IDF(q_i) = \log\frac{N - n(q_i) + 0.5}{N + 0.5} \tag{6}$$

其中$N$表示总文本数，$n(q_i)$表示出现过语素$q_i$的文本数，通过该式，我们能知道加入一个词在文本集合中出现的频率越高，则权重越低，即会拉低BM25指标。

后者对于语素$q_i$与文档$d$的相关性得分$R(q_i,d)$由下式给出：

$$R(q_i,d)=\frac{f_i*(k_1+1)}{f_i+K}*\frac{qf_i*(k_2+1)}{qf_i+k_2}$$
$$K=k_1*\left(1-b+b*\frac{dl}{avgdl}\right) \tag{7}$$

其中$k_1,k_2,b$称为调节因子，可以根据经验值进行设定，一般$k_1=2,\ b=0.75$，$f_i$表示语素$q_i$在文本$d$中出现的次数，$qf_i$表示语素$q_i$在$Query$中出现的次数，由于在本论文中语素$q_i$即为话题$t$，也就是说$qf_i$默认为1，所以综合起来，BM25指标的公式如下：

$$\mathrm{BM25(Q,d)}=\sum_{i=1}^{n}\log\frac{N-n(q_i)+0.5}{N+0.5}*\frac{f_i*(k_1+1)}{f_i+k_1*\left(1-b+b*\frac{dl}{avgdl}\right)} \tag{8}$$

## 3.4. 小颗粒文本抽取

在3.3中讨论的可迭代的优化聚类的展开过程中最根本的就是话题向量的计算，即将每个话题通过固定维度的向量在隐形空间表达出来。此处我们采用SkipGram模型[15]学习得到每个话题向量。该模型基于文本的输入，通过计算每个词与设定窗口大小内的文本同时出现的概率来计算词向量，也就是说具有相似向量的词在语义文本中具有相似的语境，或者说具有相似的上下文环境。基于SkipGram模型的方法可以有效获取不同词间的语义相似性，在NLP任务中十分常用。

对SkipGram而言，已知的是当前词$t$，需要预测的是其上下文环境，也就是我们设定的窗口大小内的文本出现概率。用公式来表示就是，基于固定的文本$D$，对于每个词$T$，假设窗口大小$W_t$，我们定义条件概率对数函数如下：

$$\mathrm{logp}(W_t|\mathrm{t})=\sum_{w\in W_t}\log p(w|t)=\sum_{w\in W_t}log\frac{v_t v'_w}{\sum_{w'\in V}v_t v'_{w'}} \tag{9}$$

其中$v_t$表示词$t$的词向量，$v'_w$表示的是窗口$W_t$内任一词$w$的词向量，$v'_{w'}$表示全体文本中任一词$w'$的词向量,$V$表示全体文本中出现的单词。那么对于全体文本的目标函数，也就是总条件概率对数函数如下：

$$L=\sum_{t\in D}\sum_{w\in W_t}\log p(w|t) \tag{10}$$

词向量的计算则可以通过梯度下降法和随机负采样[15]的方法最大化目标函数。

但是如果我们仅使用全体文本作为训练文本获得话题向量，我们发现该方法存在一个缺陷，那就是在更小颗粒度话题集合的分层聚类过程中无法有效找到话题间的区别，即无法区分出话题集合的子聚类。举个例子，如图二所示，我们发现如果两个词组在全体文本中具有相似的上下文文本，因此具有相近的空间向量，在使用全体文本用作向量训练的时候，我们能很轻易地将这两个词从根节点分出来，并划分到machine learning这一个子话题聚类中。但是，假如我们要将machine learning话题集合进行更小精度的话题分类，由于基

于全体文本训练得到的reinforcement learning和machine learning向量非常相近，所以此次话题分类会很困难，也就是说在该话题向量的对应空间内无法有效地对话题集合进行聚类。

因此小颗粒文本集合抽取就非常的重要，其目的是在于在更小精度的话题集合中更新各个话题的话题向量，便于在进行优化球形聚类的时候提高词向量间的区分度，换句话讲，就是对于任意一个话题集合$C$（不包括全体话题集合），我们使用小精度的文本抽取方法更新文本集合，从而更新话题向量。具体任务是从全体文本$D$中找到和话题集合$C$相关的文本集合$D_C$。为了获得文本集合$D_C$，我们首先利用*TF-IDF*算法计算话题集合$C$中任意一个话题$t$和全体文档集合$D$中任意一个文档$d$的相关性$tf_idf(t,d)$，该算法是一种用于资讯检索与资讯探勘的常用加权技术。作为一种统计方法，用以评估单一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。公式如下：

$$tf\_idf_{(t,d)} = tf(t,d) * idf(t,d)$$
$$tf(t,d) = \frac{f_{t,d}}{len(d)}$$
$$idf(t,d) = \log\frac{|D|}{1+|\{j: t \in d\}|} \tag{11}$$

其中$f_{t,d}$表示词$t$在文本$d$中出现的次数，$len(d)$表示文本$d$的总词数，$|D|$表示文本集合的总文本数，$|\{j: t \in d\}|$表示在文本集合中出现过词$t$的文本$d$的总数。

然后利用相关性$tf_idf(t,d)$加权，按比例将话题$t$的词向量复制给文档向量；最后通过输入话题集合$C$的几何平均话题向量查找在隐性空间中最接近的$Top\ M$个文档向量，生成小颗粒的文本集合$D_c$。基于我们获得的新的小颗粒文本集合$D_c$，利用SkipGram模型重新更新话题集合的话题向量，用于对该话题向量进行分层聚类。

# 第四章 实验开展

在实验部分，我们一共会用到两个数据集：（1）IEEE2017 数据集，其共包含了 14 千条论文记录，包括论文标题、论文摘要和论文关键词等信息，论文包含计算机视觉、计算机网络、机器学习等各大热门领域。通过利用 AutoPhrase 模型，从论文标题和论文摘要中共抽取出了 21110 个名词词组用于构建话题树；（2）DBLP 数据集共包含了 180 千条论文记录，包括论文的标题信息，论文包含信息提取、密码学、推荐系统、网页搜索等领域。通过利用 AutoPhrase 模型，从论文标题中共抽出了 50 千个名词词组用于构建话题树。

## 4.1. 实验数据获取

对于第一部分IEEE2017的数据集，我们通过爬虫技术获取了从2017年1月16日到2017年12月26日由ieeexplore收录到的共14千余条论文记录。爬虫技术主要是要解决三个问题：（1）对于网页链接的搜索策略；（2）对抓取目标文档的定义描述；（3）对网页数据的分析过滤。对于第一个问题，我们很容易可以发现ieeexplore对论文收录的方式是渐进式的，即通过编号的累加来依次往数据库中加入新的论文数据。对于第二个问题和第三个问题我

们能发现该数据库对论文的文本信息具有较高结构化的存储，同时这也大大简便了我们在对网页源码进行解构过程中对目标信息的抽取，从网页的文本源码中，我们成功抽取到了相关论文的标题、摘要、收录会议或期刊、收录时间、作者等基本信息，在本实验中我们仅采用论文标题和摘要信息。

对于第二部分DBLP数据集是由DBLP数据库官方开源的数据集，可以直接使用，在本实验中我们仅用到了其中论文标题的信息。

## 4.2. 实验模型比较

在本实验过程中，我们采用以下几种方法作为本论文使用的模型TTG的基准线：

（1） HLDA（Hierarchical Latent Dirichlet Allocation）[3]是一种分层主题模型。该模型计算了通过重复采样从根节点到叶子结点路径上话题以及话题内的单词生成的文本概率。我们采用该模型进行话题树的生成，其中每个节点都代表一个话题。

（2） TTG-K（Topic Tree Generator by Kmeans）是TTG模型的变形，即在话题集合的聚类分层过程中，将球形聚类模型该用Kmeans聚类模型。也就是说不需要通过计算向量间的余弦距离进行聚类，而是通过计算向量间的欧式距离进行聚类。

（3） NoAC是TTG模型的子步骤，即在对话题集合划分为$k$个子话题过程中，不使用优化的球形聚类模型，而仅仅采用基本的球形聚类模型。也就说无需计算话题与话题集合所对应的文本集合的代表性参数，直接根据在隐性空间内的话题向量进行聚类。

（4） NoLE也是TTG模型的子步骤，即在更小精度的话题集合递归聚类过程中，不经过小颗粒文本抽取步骤，不更新话题向量而直接进行优化的球形聚类过程。也就是说所有话题的话题向量无论在哪种精度下的聚类任务下均是通过全体文本集合的训练得到的。

## 4.3. 实验参数设定

在本实验过程中，我们对IEEE2017数据集进行四层分类，对DBLP进行四层分类。

对于TTG模型而言，我们需要设定4个参数，word2vec模型训练过程中的词向量维度s、对话题集合进行分割过程中的话题聚类个数$k$、利用TF-IDF抽取小精度文本集合的文本优先个数$m$和用于鉴别常见话题时的代表性指标阈值$\delta$。对于IEEE2017和DBLP的数据集，我们均采用$s = 128, m = 1000$进行训练，对于前者我们发现当$k = 5, \delta = 0.22$时有模型结果较好的表现，对于后者DBLP数据集我们则设定$k = 5, \delta = 0.25$。因为通过这几个参数的设定，我们发现TTG模型在优化的球形聚类过程中能较好地将常见话题和子话题集合分开，并且得到较好的子话题集合聚类。

对于HLDA模型而言，我们需要设定三个超参数：（1）在水平分布上的平滑参数$\alpha$;（2）在CRP过程的平滑参数$\gamma$;（3）关于主题词分布的平滑参数$\eta$。我们设定$\alpha = 10.0, \gamma = 1.0, \eta = 1.0$，并且基于该三个参数，我们同样利用HLDA模型对两个数据集进行话题树的构建。

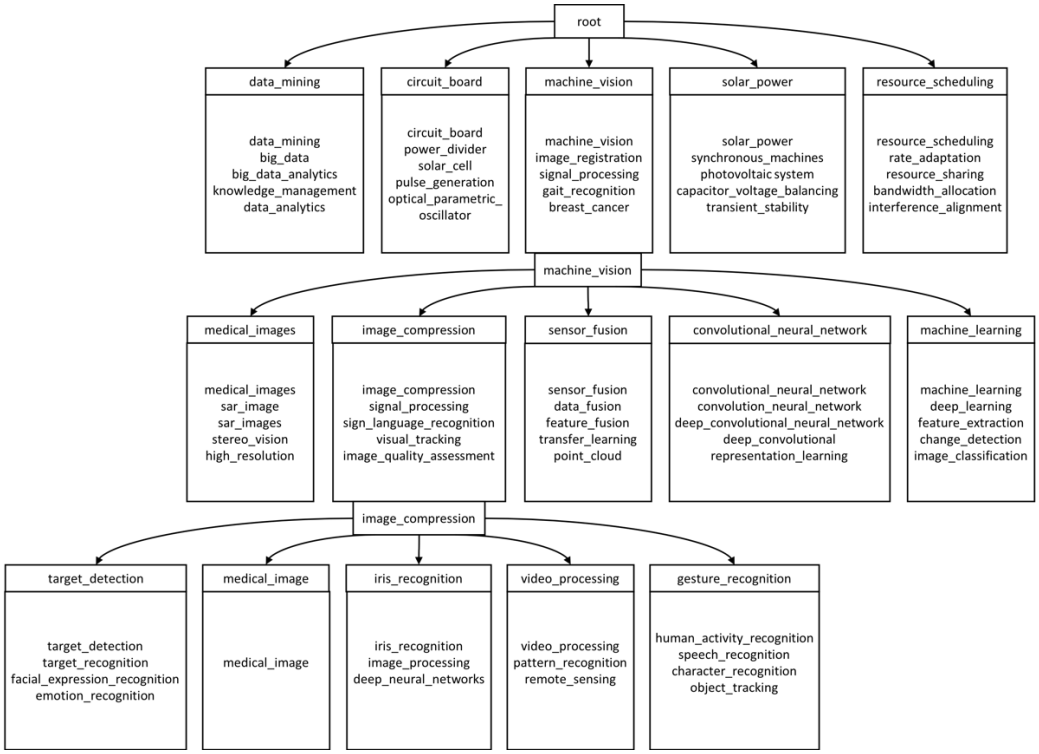
对于剩余三个模型（TTG-K、NoAC、NoLE），由于其均是TTG模型的衍生模型，所以与TTG模型拥有相同参数且设置相同取值，即$s = 128, m = 1000, k = 5, \delta = 0.25$。

## 4.4. 实验结果分析

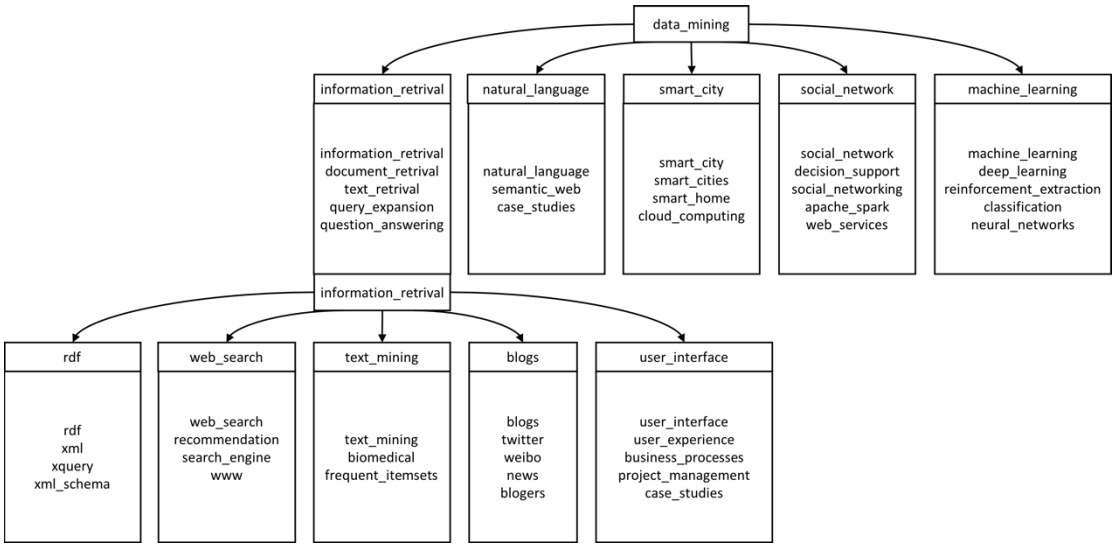
### 4.4.1. 定性化分析

在本部分我们将通过IEEE2017数据测试比较上述五个模型结果的比较。我们利用五个

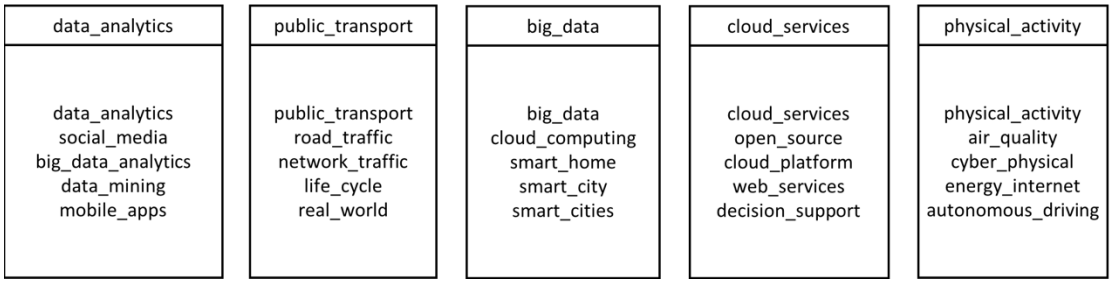
模型各自实现关于4层话题树的构建，并依次在图四至图八中展示实验结果。



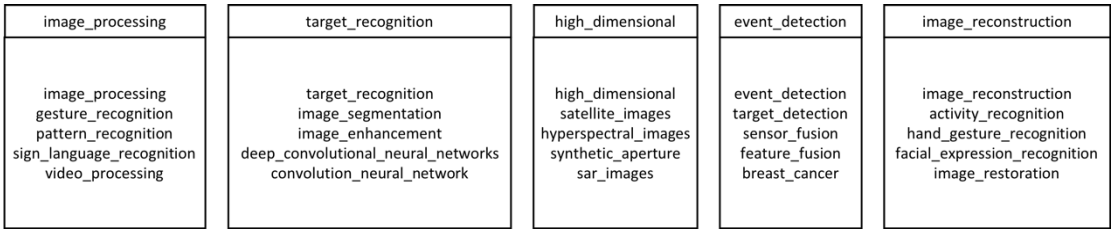

**图四　TTG模型在machine_vision领域的四层展开**



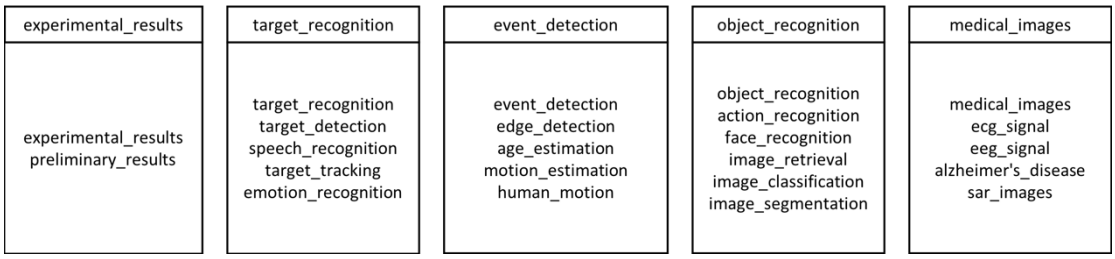

**图五　TTG模型在data_mining领域的四层展开**




**图六　TTG_k模型在data_mining领域的二至三层展开**




**图七　NoLE模型在image_compression领域的三至四层展开**




**图八　NoAC模型在image_compression领域的三至四层展开**

在本部分，我们定性的将上述几个模型的实验结果做了呈现。图四表现的是基于IEEE2017数据集，利用TTG模型对所有话题集合做了一次分层聚类，且对第二层中的machine_vision这个话题集合做了更深一步的分层聚类工作。在第二层中，我们能明显发现在初始定义聚类结果为5类的前提下，话题首先被分类为data_mining、circuit_board、machine_vision、solar_power和resource_scheduling五大类。这五个话题是通过TTG模型的优化球形模型算法得到的，且是在利用代表性参数计算结果中具有最大参数结果的话题名称。

在这五个话题聚类中，我们可以发现类别与类别之间的差距已经较为明显，但是却存在一定的不足，即话题种类并不全覆盖数据集所涵盖的话题范围，可能原因是在预先设定的聚类数目不完全精确，存在一定的偏差。另一个不足是部分话题过于确切，即不像是一个相对来说比较宽泛的词，可能的原因是该词在该话题集合对应的文本集合中出现频率较高且在兄弟话题集合对应的文本集合中出现频率较低。

在图四和图五中，我们分别基于IEEE2017数据集，采用TTG模型对machine_vision和data_mining两个二层话题集合做了进一步的分层聚类，同样也是在每一层均分为5类。在machine_vision话题集合的分层聚类过程中，我们也得到了较好的五类聚类分割，涵盖了在machine_vision方面常见的应用场景和实用方法，包含了medical_images、image_compression、sensor_fusion、convolutional_neural_network、machine_learning五个话题集合；在更下一层，我们对image_compression话题集合做了进一步的分层聚类，得到了几个具体的应用场景，包含了target_detection、medical_image、iris_recognition、video_processing、gesture_recognition。这几个应用场景都是image_compression常应用到的领域，目标探测、医学成像、精彩视频预测等都是现如今各大AI巨头公司花巨资攻关的技术难题，所以此次分类的结果是比较不错的。其次对于图五中data_mining相关话题集合的分层聚类过程中，我们得到了包括information_retrival、natural_language、smart_city、social_network、machine_learning在内的五大子话题聚类。在这一次分层中我们发现出现了不是特别相关的smart_city这一类话题，可能原因是在上一层聚类中相比于其余兄弟话题集合，smart_city相关话题和data_mining相关话题距离较近，但在此层聚类中由于小颗粒度文本集合的抽取以及词向量的重算使得smart_city相关话题和data_mining相关话题分离开来，单独聚成一类。

在图六中，我们利用TTG_k，即将球形聚类换成标准kmeans算法的方法基于IEEE2017数据集对第二层中data_mining相关话题进行了进一步的分层聚类，发现结果包含了data_analytics、public_transport、big_data、cloud_servises、physical_activity这五大聚类结果，很明显一点就是该方法得到的聚类直接看来与data_mining领域的相关性并不是很高，所以kmeans算法在此处对于话题向量的聚类任务上表现并不是非常好。

与此同时，我们也在图七和图八中进一步对machine_vision话题集合利用NoLE和NoAC两个模型进行了话题集合的分层聚类任务。我们发现基于NoLE的分类方法会使得某些子话题集合中的集合应该属于该子话题集合的兄弟话题集合。因为NoLE的分类方法中，话题集合没有再次利用tf-idf算法重新生成小颗粒度的文本集合，也就是说并没有在小颗粒度的话题分层任务中重新计算话题的词向量。而NoAC模型因为缺少了计算代表性指标这一项任务，所以其只能实现最基本的聚类，而无法实现分层的效果，也就是说并无法判断一个话题集合中的常见话题和子话题集合。

4.4.2. 定量化分析

在本部分我们将定量化描述各方法得到的结果的效果。对话题树构建好坏的评价是一件非常困难的事情，因为并不存在所谓的ground truth，并且对话题树构建的质量评价需要考虑到多方面因素。我们在评价过程中考虑到以下几方面因素：

（1）关系准确性：旨在衡量给定分类中是否真具有准确的上下层关系。

（2）术语一致性：旨在量化一个主题的最高层术语在语义上具有一致性。

我们将以上三个方面的评估通过以下方法定量计算如下。首先，对于关系准确性，我们将所有话题树中的父子对分类用户调研来判断这些父子对的准确性。在用户调研阶段，我们邀请了10位学生作为评估人员。对于每个父子对，我们向至少三位评估者展示父子话题（每个父子对由五个代表性词语构成），并询问给定对是否有效的亲子关系。收集评估人员的答案后，我们只需计算评估人员的投票比例就可以得到关系准确性的定量化指标。

其次，为了定量化描述术语一致性，提供给评估人员某个主题代表性指标最高的前五名，并这些术语中注入了一个从兄弟主题集合中随机选择的非该主题集合中的词。随后，我们向评估者展示这六个术语，并询问哪一个是不属于这一主题集合的术语。直观地说，话题集合的一致越高，评估者就越有可能正确识别出不属于该主题集合的词语，因此我们计算正确评价的比例作为术语一致性评分。

**表1 模型结果指标评价**

| | 关系准确性 | | 术语一致性 | |
|---|---|---|---|---|
| 模型 | IEEE | DBLP | IEEE | DBLP |
| HLDA | 0.27 | 0.32 | 0.43 | 0.37 |
| NoAC | 0.35 | 0.37 | 0.37 | 0.45 |
| NoLE | 0.37 | 0.41 | 0.53 | 0.43 |
| TTG_k | 0.45 | 0.52 | 0.62 | 0.61 |
| TTG | **0.58** | **0.67** | **0.71** | **0.69** |

表1主要通过用户调研的形式统计了四个模型结果的关系准确性和术语一致性特征，前者主要是为了衡量在给定的分类中上下层之间是否具有准确的父子关系，即该子话题集合是否真的属于该父话题集合；后者主要是为了量化一个话题集合内的话题具有一致性的语

义，即真的是彼此相关的话题才应该聚到一起。根据结果我们能明显发现HLDA和NoLE的表现较差，而TTG模型在IEEE2017数据集和DBLP数据集中均具有较好的表现，尤其是在同时包含了论文标题和论文摘要的IEEE2017数据集中表现更佳。

对该结果，我们认为原因是多方面的。首先HLDA模型对文档主题和主题词分布做出更强的假设，这可能不适合用于真实数据。 其次，主题建模方法的代表性术语是纯粹基于话题各自的分布，而基于词向量的方法则对于具有代表性的话题具有较强的独立性。

# 第五章 论文总结

在本论文中我们研究了从非结构化文本中实现话题树的构建的问题。本文采用的TTG模型(Topic Tree Generator)实现了高精度且自动化无需人为先验知识输入的话题树构建过程。该模型主要将各话题放入了隐性空间计算语义向量，并利用分层聚类的算法自顶向下递归构建话题树。它由一个优化的球形聚类模块组成，该模块在分割不同颗粒度的主题时将话题集合分配到适当的层级，以及包括一个基于小精度文本集合训练得到的词向量模块。该模块可以在不同颗粒度的分类层级中实现词向量的训练。在我们的实验中，我们证明了这两个模块都有助于提高话题树的质量，这使得TTG模型比现有的在话题树构建任务过程中其他的方法更具优势。

当前版本的TTG模型的一个局限性是，在做优化的球形聚类模型下，它需要预先指定聚类数目。由于聚类个数是事先确定的并且针对每个主题都是固定的，因此该机制在实践中可能不会产生最佳结果。将来，一个改进空间在于让TTG模型在不同分类等级可以允许它自动确定聚类过程中最佳的聚类个数，如利用Elbow Algorithm寻找拐点的办法定义相对最佳聚类数目。针对这个问题的方法包括基于非参数模型的聚类技术和将相对轻量的用户交互输入融入模型中的全新机制。

# 参考文献


[1] E. AgichteinandL. Gravano. Snowball:Extractingrelations from large plain-text collections. In ACM DL, 2000.

[2] S. Brin. Extracting patterns and relations from the world wide web. In International Workshop on The World Wide Web and Databases, pages 172–183, 1998.

[3] A.Carlson,J.Betteridge,B.Kisiel,B.Settles,E.R.HruschkaJr,andT.M.Mitchell. Toward an architecture for never-ending language learning. In AAAI, volume 5, page 3, 2010.

[4] B. Cui, J. Yao, G. Cong, and Y. Huang. Evolutionary taxonomy construction from dynamic tag space. In WISE, pages 105–119, 2010.

[5] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. Machine Learning, 42(1/2):143–175, 2001.

[6] R.Fu,J.Guo,B.Qin,W.Che,H.Wang,andT.Liu.Learningsemantichierarchies via word embeddings. In ACL, pages 1199–1209, 2014.

[7] G. Grefenstette. Inriasac: Simple hypernym extraction methods. In SemEval@NAACL-HLT, 2015.

[8] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In COLING, pages 539–545, 1992.

[9] M. Jiang, J. Shang, T. Cassidy, X. Ren, L. M. Kaplan, T. P. Hanratty, and J. Han. Metapad: Meta pattern discovery from massive text corpora. In KDD, 2017.

[10] Z. Kozareva and E. H. Hovy. A semi-supervised method to learn and construct taxonomies using the web. In ACL, pages 1110–1118, 2010.

[11] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. On semi-automated web taxonomy construction. In WebDB, pages 91–96, 2001.

[12] X. Liu, Y. Song, S. Liu, and H. Wang. Automatic taxonomy construction from keywords. In KDD, pages 1433–1441, 2012.

[13] A. T. Luu, J. Kim, and S. Ng. Taxonomy construction using syntactic contextual evidence. In EMNLP, pages 810–819, 2014.

[14] A. T. Luu, Y. Tay, S. C. Hui, and S. Ng. Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In EMNLP, pages 403–413, 2016.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In NIPS, pages 3111–3119, 2013.

[16] N. Nakashole, G. Weikum, and F. Suchanek. Patty: A taxonomy of relational patterns with semantic types. In EMNLP, pages 1135–1145, 2012.

[17] A. Panchenko, S. Faralli, E. Ruppert, S. Remus, H. Naets, C. Fairon, S. P. Ponzetto, and C. Biemann. Taxi at semeval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. In SemEval@NAACL-HLT, 2016.

[18] S. P. Ponzetto and M. Strube. Deriving a large-scale taxonomy from wikipedia. In AAAI, 2007.

[19] J.Seitner,C.Bizer,K.Eckert,S.Faralli,R.Meusel,H.Paulheim,andS.P.Ponzetto. A large database of hypernymy relations extracted from the web. In LREC, 2016.

[20] W.Wu,H.Li,H.Wang,andK.Q.Zhu.Probase:Aprobabilistictaxonomyfortext understanding. In

Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pages 481–492. ACM, 2012.

[21] H. Yang and J. Callan. A metric-based framework for automatic taxonomy induction. In ACL, pages 271–279, 2009.

[22] Y. Zhang, A. Ahmed, V. Josifovski, and A. J. Smola. Taxonomy discovery for personalized recommendation. In WSDM, 2014.

[23] J.Zhu,Z.Nie,X.Liu,B.Zhang,andJ.-R.Wen.Statsnowball:astatisticalapproach to extracting entity relationships. In WWW, 2009.

[24] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, Jiawei Han, "Automated Phrase Mining from Massive Text Corpora", accepted by IEEE Transactions on Knowledge and Data Engineering, Feb. 2018.

# 谢辞

将基于海量摘要文本的话题树构建这个内容作为我的毕业论文其实是一件非常具有挑战的任务，因为其本身在文本挖掘中信息抽取这个子类下一个非常新颖的任务，并且没有一个非常完备的方法论可供参考，但是这正体现了这个任务的价值，因为一旦能够较好地解决话题树构建这个任务，能够构建出一个无需过多先验知识就可以跑通的模型，这是一件非常有实际价值的事情，所以我很荣幸能够主导这个项目，从论文调研和代码构建过程中收获到了很多。

在这里，我想感谢给了我这个课题灵感的老师和实验室的同学们，感谢大家在我论文搜集、代码书写和实验开展过程中提供的无私帮助。同时，我也非常感谢伊利诺伊香槟分校张学长和陶学长两位在数据集和模型说明方面给我提供的帮助。

虽然这个项目只是我毕业设计的一个课题，但是我相信在未来的学习研究过程中，我依旧会对这个课题以及其衍生的相关课题进行更深入的调研和学习!感谢毕业设计给我这个机会进一步熟悉这个领域的发展以及对我研究兴趣的激发，希望自己在未来能有更大的进步!

# TTG: TOPIC TREE GENERATOR BY ANLYZING ABUNDANT PAPER TITLE AND ABSTRACT

In this paper, we utilize a unique and excellent model called TTG (Topic Tree Generator) to construct a hierarchical topic tree. It is not only a very basic task in corpus analysis, but also plays a great important role in information filtering, recommendation systems, search engines, .etc.

According to pre-research on this field, we found the existing methods are based on three methods, pattern-based methods, clustering-based methods and supervised methods. The first way is to classify the organizational components of sentences by extracting contextual semantics. However, by considering each phrase as an independent concept node, the topic proximity and semantic relevance will be more emphasized. And the second idea focus on finding related similarity of specific feature of terms in latent space, like calculating the distance between term embedding. And the last idea is under supervision and requires prepared knowledge database by abundant experts to classify the relationship between several terms.

In this article, we use a TTG (Topic Tree Generator), a multi-level topic-building classification method, in which each node represents a conceptual topic and a cluster of semantically related topics.

The description of the problem is to construct the topic tree, and the input includes two parts:(1)All documents $D$; (2)All topics $T$. The group of topics $T$ are extracted from documents $D$ by AutoPhrase method, which represents the whole terms for constructing topic tree. Based on the documents $D$ and topics $T$, we hope to construct a tree-structured hierarchy $H$. The each node $C$ in $H$ represents a set of terms $T_C$ with coherent semantical feature. Assuming the sub-nodes of $C$ is $S_C = \{S_1, S_2 \ldots S_n\}$, each $S_i$ is equal to one sub-topic group of node $C$.

As described before, the first task is to extract meaningful phrases for non-structural documents. AutoPhrase method is implemented in this paper, which is an automated phrase mining method to extract quality phrases form a large collection of documents without human labeling effort, and with only limited, shallow linguistic analysis. The main input to the automated phrase mining is a textual word sequence in a particular language and a specific domain, of arbitrary length. The output is a ranked list of phrases with decreasing quality. The AutoPhrase framework is combined by robust positive-only distant training and POS-guided phrasal segmentation module, and the quality phrase should meet following criteria: popularity, concordance, informativeness and completeness. In our paper, the input positive label pools are phrases extracted from Wikipedia and all related keywords labelled by ieeexplore.

When we begin to design this model, two challenges need to be addressed. First, it is significant to determine the proper levels for different topic. When splitting a group of topic nodes into lower level, not all topics should be pushed down to the child level. For instance, the general topics ‘computer_science’ and ‘cs’ should absolutely stay in the root node instead of being allocated into any child nodes for constructing the cs related topic tree. Thus, it is problematic to distinguish the general topics from topic groups and allocate other topics into child level. Second, because this

model is based on term embedding, and global embedding has limited discriminative power at lower level. Term embedding are typically learned by collecting the context evidence from the corpus, such that terms share similar contexts tend to have close embedding. For example, because term embedding won't be updated in the whole recursive process, when we splitting the computer vision topic, we find the term 'computer_vision' and 'computer_graphics' have close global embedding, and it is hard to discover quality sub-topics for the computer vision topic.

To solve these two difficulties, we proposed a classification method that uses word vectors and hierarchical clustering to accomplish multi-level topic construction in a recursive process. In the recursive process, we have adopted: (1) an optimized spherical clustering model that can ensure that certain word combinations are shifted downwards when subject clustering is layered to smaller grained topics; (2) A local term embedding improves the accuracy of the model when stratified to a smaller granularity theme by training word vectors under a smaller granularity of text.

Specifically, the adaptive clustering module is designed to split topic group into general topics and several sub-topics. It combines the basic spherical k-means algorithm, which makes clustering via calculating cosine distances, and representativeness between one topics and related documents group. The key idea is to iteratively identify general terms and refine the sub-topics after pushing general terms back to the parent. For measuring term representativeness of related documents, we address that a representative term should appear frequently in the related documents but not in the sibling documents. For these two characters, we use the popularity and concentration separately. And the representativeness is the square root of popularity and concentration. Tf-idf algorithm is implemented in the popularity and BM25 relevance is adopted in the concentration. In information retrieval, tf–idf, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Tf-idf is one of the most popular term-weighting schemes today. Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. And BM25 relevance is a ranking function used by search engines to rank matching documents according to their relevance to a given search query. It is based on the probabilistic retrieval framework developed in the 1970s and 1980s by Stephen E. Robertson, Karen Spärck Jones, and others.

The local embedding process aims to update term embedding in the recursive clustering process. Based on SkipGram model for learning term embedding, it can calculate the relationship between a term and its context terms in a sliding window, such that the terms share same context tend to have close embedding. Local embedding module helps to enhance the discriminative power of term embedding at lower level. For any topic which is not the root topic, we learn term embedding by sub-topic related documents. Here we first compute the document embedding by using tf-idf weighted average of the term embedding related. Based on the document embedding, we use the mean direction of sub-topics as a query vector to retrieve the top closest documents and form the sub-topics related documents to update the sub-topic embedding.

In the experiment section, we proposed quantitative and qualitative method to compare the strengths and weaknesses between HLDA, TTG, TTG-k, NoAC and NoLE models. As for these

five models, the first is a nonparametric hierarchical topic model, models the probability of generating a document as choosing a path from the root to a leaf and sampling words along the path. And the last three models are variant of TTG model, separately utilize k-means algorithm instead of spherical k-means algorithm, without the adaptive clustering module and without local embedding process. In addition, we implement user study to quantitatively evaluate the quality of the constructed topical tree by considering relation accuracy and term coherency factors, which aims at measuring the portions of the true positive parent-child relations in a given tree and quantifying how semantically coherent the top terms are for a topic.

In conclusion, we study the problem of constructing topic tree from a give group of documents. The model TTG combines the term embedding and adaptive spherical k-means techniques to construct the topic tree in a recursive way. The local term embedding based on specific relationship between documents and terms via tf-idf principle is beneficial for discover more quality sub-topics and general topics for one topic. And the adaptive spherical k-means algorithm helps to learn term embedding to maintain strong discriminative power at lower classification levels.

However, the limitation of this model is clear. The model requires some pre-defined parameters, like specific number of clustering, the threshold of discriminating general topics and sub-topics, .etc. Since these parameters are different from dataset to dataset, the choice of these parameters is a boring and dirty work for many people. In the future, the elbow algorithm or some human interactive mechanism should be added into the model, in other words, to allow it to automatically determine the optimal number of clustering of each parent topics in the recursive process.