

清 华 大 学

综 合 论 文 训 练

题目：基于在线商品评论的对象评价
体系构建

系 别：电子工程系

专 业：电子信息科学与技术

姓 名：王 颖

指导教师：邓北星 教授

2017年05月25日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名：_____导师签名：_____日 期：_____

中文摘要

本文提出一种基于自由评论文本的、跨领域通用的层次化树状商品评价体系（属性树）的构建方法。该方法有别于现有的其他方法，同时具备不依赖领域知识库、不依赖结构化或半结构化文本、评价体系层次化、精确度较高的特点。本文主要创新点在于，提出抽取属性间的从属关系并用于属性词聚类标签的提取，并提出若干自定义规则对传统二分类方法得到的属性树进行校正、完善和修剪，使得属性树的整体树形结构和准确率都有较大提高。

关键词：层次化聚类；商品属性抽取；情感计

ABSTRACT

This paper proposed a new way to construct a domain-independence hierarchical evaluation system (the aspect tree) for products based on free text from online customer reviews. Our method is different from other existing methods, as it is an accurate hierarchical system which does not need any knowledge base or any structured or semi-structured text. The contribution of this paper is to extract the subordinate relationship between aspects and use it to compute the cluster labels after clustering aspects. And we propose several rules to correct, perfect and prune the aspect tree obtained by traditional two classification methods to get a more reasonable aspect tree with higher accuracy.

Keywords: hierarchical clustering; aspect tree extraction; sentiment analyze

第 1 章	引言	1
1.1	研究背景	1
1.2	本文主要工作概述	1
1.3	论文安排	1
1.3.1	背景介绍	1
1.3.2	商品属性树	2
1.3.3	结果及评价	2
第 2 章	研究现状	3
2.1	已有工作概述	3
2.2	与已有工作对比	4
第 3 章	属性树构建概述	5
3.1	属性树概述	5
3.2	任务难点及挑战	5
3.3	算法流程	5
第 4 章	商品属性抽取	7
第 5 章	属性关系抽取	8
5.1	从属关系定义	8
5.2	抽取规则	8
5.3	抽取特点	9
第 6 章	初始属性树构建	10
6.1	概述	10
6.2	属性词向量化	10
6.3	向量预处理	11
6.4	自适应二分聚类	12
6.5	类别标签提取	13
6.6	小结	15
第 7 章	最终属性树构建	16
7.1	初始属性树中存在的问题	16
7.2	属性树的校正、完善与修剪	16
7.3	属性树各结点权重计算	22

7.4	小结	23
第 8 章	实验结果及评价	25
8.1	实验数据集介绍	25
8.2	属性树结果展示	25
8.3	属性树结果分析	30
8.3.1	评价指标	30
8.3.2	整体结果分析	30
8.3.3	初始属性树对比 Shi B 的属性树	31
8.3.4	最终属性树对比初始属性树	31
8.4	小结	33
第 9 章	总结与展望	34
插图索引	35
表格索引	36
参考文献	37
致 谢	38
声 明	39
附录 A	书面翻译	40

第1章 引言

1.1 研究背景

随着信息产业的迅猛发展，网络购物日益普及。但网上购物的过程中，消费者在不能实际接触商品本身的情况下，常常需要借助他人对商品的评论来判断一件商品的优劣，从而做出正确抉择。然而一些热门商品的评论动辄成百上千条，人工逐条浏览耗时耗力，使用程序算法基于评论文本进行情感分析，可以辅助消费者快速决策，节省时间精力。不过，不同消费者对商品的需求不同，所关注的方面也不同。例如，同样是购买智能手机，有的消费者爱看影视作品，会更关注手机的屏幕尺寸、分辨率、色域等，而有的消费者出于商务旅行的需要，会更在意手机的电池续航能力。因此在进行情感分析之前需要构建一个合理的评价体系，该评价体系应涵盖商品的各个层次、各个方面。

1.2 本文主要工作概述

本文在前人工作基础上，从网站上爬取大量商品评论自由文本^①，自动从中抽取出商品的各个属性词，抽取属性词之间的从属关系，并对属性词进行聚类得到层次化的属性树，最后对该层次化属性树进行校正及修剪，得到最终结果。本文的主要贡献在于，提出属性间从属关系的抽取方法，利用从属关系改善属性树中各类属性词的标签，并提出校正、完善、修剪属性树的若干方法，能够不依赖领域知识库和文本结构自动生成一棵较高精确度的属性树。

1.3 论文安排

本文共分为三部分，具体如下。

1.3.1 背景介绍

第一部分背景介绍，包括第一章和第二章。其中第1章引言，简要介绍了研

^① 本次实验所用的评论文本爬取自美国亚马逊购物网站（<https://www.amazon.com>）。

究背景及意义和论文的主要工作。第2章研究背景，介绍了关于商品评价体系的构建的几种主要方法和其各自优缺点，以及这些方法与本文方法的联系及区别。

1.3.2 商品属性树

第二部分是商品属性树的构建方法，是本文的重点，包括第3到7章。共分为三个层次来介绍。

第一层次，是第3章“属性树抽取概述”。阐述了本文构建商品层次化评价体系——属性树的基本流程框架，及其中所涉及到的难点和存在的挑战，以及本文的解决方案。

第二层次，是第4章和第5章，从属性及其关系的层次展开。第4章“商品属性抽取”，介绍了在前人方法基础上从没有任何结构信息的自由评论文本中抽取商品的属性词汇的方法。第5章“属性关系抽取”介绍了如何在没有相关领域知识库等先验知识的情况下，从自由评论文本中抽取属性词之间的关系。

第三层次，是第6章和第7章，从属性树这一整体层次展开。第6章“初始属性树构建”详细介绍了从零散的属性词到整体属性树的过程，并对每一步骤的方法进行了分析。第七章“最终属性树构建”说明了初始属性树中存在的不足，并提出若干改进方法，得到完善的最终属性树。

1.3.3 结果及评价

该部分是对上一部分“商品属性树”的评价，包括第8章和第9章。

第8章“实验结果及评价”，介绍了本文实验所用的数据集及其特点，展示了最终属性树的结果，并将本文得到的初始属性树、最终属性树与前人工作做了对比，展示了属性树准确性逐步提高的过程，论证了本文方法的有效性。

第9章“总结与展望”，总结了本文的主要贡献与创新，包括：提出一种基于自由评论文本的、领域通用的、准确的层次结构评价体系构建方法；提出抽取属性间的从属关系，并利用它来修正初始属性树的标签；提出若干自定义规则校正、完善和修剪属性树，使其层次结构更加合理，并提高其总体精确度。该章还列举了可能的应用场景，包括情感计算、数据挖掘等。

第2章 研究现状

2.1 已有工作概述

对一件商品的评价，包含方方面面的评价。其中有对它整体的评价，也有对它的各个属性的评价。而想要细粒度地挖掘商品各方面信息，就需要从属性层面入手，提取出评论者对商品的各个属性的评价态度，因此大多数评价体系都是基于商品属性的评价体系。而现有的评价体系构建方法分为两大类，一类是侧重于商品属性的抽取，将对商品的评价转化为对属性的评价。另一类是侧重于层次化结构，即属性树的构建。

第一类方法中以 Liu B 等人的工作《Mining opinion features in customer reviews》^[1]为代表。该类方法从评论文本中抽取商品的各个属性，并按这些属性出现的频率高低排序，这样的评价体系优点在于可以挖掘出一些较为细致、提及较少的属性及其关联的评论语句。然而，这样的评价体系缺点也很明显——结构不合理，所有属性依次排列，缺乏层次。这直接导致属性之间的关联性在该体系中被抹去。例如，以智能手机为例，“分辨率”作为“屏幕”这一属性的子属性，在该评价体系中完全分开、独立看待。

第二类方法与第一类不同，侧重于构建层次化结构。然而由于层次化结构的复杂性及其构建难度，该类方法中大都都需要一些先验知识。例如，Yu J 等人的工作《Domain-Assisted Product Aspect Hierarchy Generation: Towards Hierarchical Organization of Unstructured Consumer Reviews》^[2]使用了树状评价体系，即商品的属性树。属性树的父子结点之间具有从属关系，子结点是父结点的从属性，例如，对于智能手机而言，“分辨率”和“尺寸”这两个属性就是“屏幕”的子属性，它们表征了“屏幕”这一属性的更细致的特征。该方法通过领域知识库的先验知识，得到一棵十分粗糙的初始属性树，再通过巧妙的转化，将各个抽取出的属性词加入初始树的过程转化成函数优化问题，通过最小化属性树整体误差来得到最终属性树。该方法的优点在于，体系层次结构清楚，能够看出各个属性之间的关系。然而该方法在构建属性树时需要借助领域知识库，由于不是每一类商品都有对应领域知识库，因此该方法不具有很好的跨领域通用性。此外，该方法得到的最终属性树的准确性，很大程度上依赖于从领域知识库中导出的初始树的合理性，若初始树不够合理，最终结果也会受到较大影响。

注重层次结构的方法中还有一种方法，以 Shi B 等人的工作《Generating a

concept hierarchy for sentiment analysis》^[3]为代表。该方法通常是通过对属性词向量化，再进行层次化聚类来得到属性树。这种方法同样能得到层次化结构的属性树，优点在于不需要领域知识库作为先验，具有很好的通用性，但该方法构建的属性树固定为二叉树，结构不合理，且属性树的准确性较差，一些不具备从属关系的结点也作为父子结点出现。

2.2 与已有工作对比

文献[1]所代表的这类工作，挖掘出较多属性，但缺乏层次结构，忽略了属性间的关联性。这类方法与本文有着显著不同，本文的方法构建的是树状的评价体系，层次结构更为丰富，属性之间的相互关联能清楚呈现。但这类方法对于商品属性的高效抽取方法为本文所借鉴。

文献[2]所代表的这类工作，体系层次结构清楚，保留了各个属性之间的关联。然而该方法需要相关领域知识库作为先验，不具有跨领域通用性。这类方法与本文也有着显著不同，本文的方法基于自由评论文本，不需要结构化或半结构化的评论语句，也不需要领域知识库作为先验，是更为通用的方法。但是Yu J等人对属性树的构建不是一步完成，而是由初始树逐步优化得到最终的属性树，这一思想本文中也有体现。本文先构建出较为粗糙的初始属性树，再通过一系列精细方法得到最终属性树。

文献[3]所代表的这类工作，不需要领域知识库作为先验，具有很好的通用性，但该方法构建的属性树结构不合理，且属性树的准确性较差。这类方法与本文较为相似，也是本文实验结果对比的基准。不同之处在于，本文抽取并应用了属性间的从属关系，并对属性树进行了校正等，相比这下，属性树的树形结构更为合理，准确度更高。

纵观现有的基于评论文本的商品评价体系构建方法，或者缺乏层次结构，如文献[1]；或者需要借助领域知识库作为先验，如文献[2][4]；或者需要文本是半结构化的，如文献[5]；或者属性树结构不合理、准确度不够高，如文献[3]。针对这一问题，本文在前人工作基础上，提出一种基于自由评论文本的、领域通用的、准确的层次结构评价体系构建方法。

第3章 属性树构建概述

3.1 属性树概述

商品的层次结构评价体系，即商品的属性树。属性树是由商品各个不同层次的属性构成的一棵深度、广度不定的树。属性树的根结点是商品自身，每个结点的孩子结点，从语义上来说，是其父结点的从属性，即子结点是对父结点更细致特征的描述。例如，“屏幕”、“相机”、“电池”等是商品“手机”的从属性，在属性树中，“手机”是根结点，“屏幕”、“相机”、“电池”等是第二层的结点。而“分辨率”、“尺寸”等是“屏幕”的从属性，是对“屏幕”更细致特征的描述，因此它们是“屏幕”的子结点，如此类推。

3.2 任务难点及挑战

为充分考虑属性树的通用性，属性树的构建算法应基于自由文本（多数购物网站的商品评论文本是自由文本，没有任何结构化的信息），且应不必依赖任何先验信息（如领域知识库）。

基于自由评论文本，相比基于半结构化和结构化文本的方法，没有任何先验知识可言，这将导致抽取出的属性词准确率受到较大影响。对此，本文使用了准确度、灵活性尽可能高的属性抽取方法，并在构建属性树后采取合理的机制滤除不合理的属性词。

而不依赖先验信息，导致没有初始的层次结构框架可以依附，必须自动生成层次结构框架，这也带来了不少挑战。依靠层次聚类算法生成的层次结构在结构和精确度上都不够可靠。对此，本文将层次聚类算法生成的层次结构作为初始框架，应用一系列精细的规则优化初始的层次结构，改善属性树的树形结构，从而得到可靠的属性树。

3.3 算法流程

为从自由评论文本中，不依赖领域知识库构建一棵高精度的商品属性树，本文将属性树构建分为以下步骤：

1. 商品属性抽取
2. 属性之间从属关系抽取
3. 属性聚类生成初始属性树
4. 属性树的完善与修剪

具体地，第 4 章阐述了从商品评论语料库中抽取商品属性的方法，先利用自然语言处理中的词性分析、句法依存关系分析处理评论文本，使用 Liu B 等人提出的“double propagation”（见文献[6]）方法，抽取商品属性，如对于智能手机的评论文本，抽取“屏幕”、“相机”、“电池”、“存储”、“分辨率”、“尺寸”等等属性。第 5 章说明了抽取属性之间从属关系的方法，具体的，依据语法依存关系，自定义依存规则，从评论文本中抽取属性之间的从属关系，得到形如“屏幕-尺寸”、“相机-像素”这样“主-从”属性对。第六章将属性词按照特定规则向量化，进行预处理后自适应二分聚类，并依据前一步提取的从属关系为每一类属性提取标签，由此生成一棵二叉初始属性树。第七章依据从属关系及若干自定义的规则，对属性树进行校正、完善和修剪，得到最终的属性树。

与 Shi B 等人的方法相比，同为基于自由评论文本、领域通用的方法，本文得到的属性树在树的结构和准确性上均有提高，详见第 8 章“实验结果及评价”。

第4章 商品属性抽取

商品属性一般为名词或名词短语,在评论文本中,属性词常常与带有情感色彩的情感词(通常为形容词)共同出现,并被情感词所修饰。因此,属性抽取需要借助词性分析和语法分析。本文实验所用的词性分析和语法分析工具为 The Stanford Parser (见文献[7])。属性抽取的方法借鉴 Bing Liu 等人的“double propagation”(见文献[6])方法,由初始情感词典(见文献[8])开始,启发式提取属性词和情感词,但在抽取规则上略微修改,以避免产生过量噪声。具体算法如下:

记情感词集合为 O ,属性词集合为 T 。初始情感词典中含有大量带有情感色彩的情感词(多为形容词),记初始词典中情感词集合为 O_0

0. 令 $O = O_0$, $T = \Phi$

1. 根据句法依存关系,查找被 O 中的情感词修饰的名词,将它们加入属性词集合 T
2. 根据句法依存关系,查找与 T 中的属性词具有并列关系的名词,将它们加入属性词集合 T
3. 根据句法依存关系,查找修饰 T 中的属性词的形容词,将它们加入情感词集合 O
4. 根据句法依存关系,查找与 O 中的情感词具有并列关系的形容词,将它们加入情感词集合 O
5. 重复步骤 1-4,直至集合 O 和集合 T 都不再变化,取集合 T 中出现频率最高的 N 个名词作为商品属性词(实验中 N 取 50)

核心步骤 1-4 用图解表示如图 2.1。



图 2.1 “double propagation” 核心步骤图解

该算法高效、灵活,可以从评论语料库中获得商品的属性词,用于属性树的构建。

第5章 属性关系抽取

5.1 从属关系定义

属性之间的从属关系，是指两个属性之间，具有的上下级关系，下级属性是上级属性的某一具体特征。例如“分辨率”和“屏幕”这两个属性就具备从属关系，其中“分辨率”是下级从属性，“屏幕”是上级主属性。主属性、从属性是相对而言的，例如，在从属关系“屏幕-分辨率”中，“屏幕”是主属性，而在从属关系“手机-屏幕”中，“屏幕”则是从属性。

在商品的属性树中，应该满足父结点是子结点的主属性，子结点是父结点的从属性。

5.2 抽取规则

通过观察评论语料，发现一对“主-从”属性常常以以下方式共同出现：

1. “A of B”形式，A、B 是两个商品属性，以这种形式出现时，A 可能是 B 的从属性，例如“size of screen”、“resolution of screen”、“quality of this phone”等等。
2. “AB”形式，A、B 是两个商品属性，以这种形式出现时，A 可能是 B 的主属性，例如“battery life”、“screen size”等等。

而在 Stanford Parser 中，句法依赖关系以三元组<依赖词，中心词，依赖关系>的形式呈现，依赖词通过特定的关系依赖于中心词，根据上述对评论语料的观察，本文对属性从属关系的提取主要依据两种语法关系，具体如下：

1. 在某一句法依赖关系中，若依赖词与中心词均为第 4 章中抽取出的属性词，且依赖关系为名词性修饰中的“of”类型（在 Stanford Parser 中具体为“nmod: of”依赖型），则认为中心词是依赖词的主属性。
2. 在某一句法依赖关系中，若依赖词与中心词的词性均为第 4 章中抽取出的属性词，且依赖关系为复合关系（在 Stanford Parser 中具体为“compound”依赖型），则认为中心词是依赖词的从属性。

5.3 抽取特点

根据这样的规则抽取出的从属关系中，一个属性词可以有多个从属性，且一个属性词也可以有多个主属性。例如“screen”的从属性有“size”、“resolution”等等，而“quality”的主属性有“phone”、“battery”等等。

然而依据这样的语法关系抽取的从属关系，存在一定错误，主要原因在于，部分语句中，无法简单根据句法依赖关系判断谁是主属性、谁是从属性。例如“screen”和“size”，“size”是从属性，“screen”是主属性，然而不论在“A of B”形式中还是在“A B”形式中，“screen”和“size”均是既可以作为A出现又可以作为B出现，如“a nice size screen”、“nice screen size”和“screen of this size”、“size of the screen”。对于这样的情况，根据句法依赖关系无法判定从属关系。

提取的从属关系整体上较为可靠，但精确度较低，存在大量噪声，正因如此，直接利用从属关系构建属性树是不可靠的，所以本文采用聚类方法生成属性树，而使用从属关系的统计特征辅助属性树的修剪完善，详见后文。

第6章 初始属性树构建

6.1 概述

使用聚类生成层次结构是常用的方法，英文称作“Hierarchical clustering”（见文献[9]）。因此本文用向量聚类的方法来生成初始属性树。具体分为属性词向量化、向量预处理、自适应二分聚类、类别标签选取几个部分，6.2 属性词的向量化一节，对比了常见几种词汇向量化方法，说明选择语境向量的原因，并阐述了语境向量的提取方法。6.3 向量预处理一节，说明了对提取出的语境向量进行的预处理方法。6.4 自适应二分聚类一节，讲述如何将已经过预处理的语境向量，自顶向下自适应二分生成一棵深度不定的二叉树。6.5 类别标签提取一节，为二叉树的非叶子结点标注名称，说明了 Shi B 等人的方法中存在的不足，并基于第 5 章的属性从属关系提出一种新的方法，阐述了新方法的原理及算法特点。

6.2 属性词向量化

词语的向量化有多种方法，最简单常用的一种叫“one-hot representation”，即对 N 个单词进行向量化，则将每个单词表示为一个 N 维向量，任意一个 N 维向量只有 1 维是 1，其余全是 0，任意两个词向量正交。即第 n 个单词的词向量，只有第 n 维为 1，其余 $N-1$ 维皆为 0。这样的表示虽然简单，但对于本文的属性词聚类不适用，因为这样的表示方法丢失了属性之间的上下文关联信息，任意两个属性词的词向量之间均无任何相似之处，聚类时计算向量间的距离也没有意义。

另一种现较为流行的词语向量化方法，是基于神经网络训练分布式词向量，以谷歌的 word2vec 为代表（见文献[10]）。这样的向量化方法将每个单词映射到一个较低维度的向量，向量每一维是一个浮点数。这样的词向量可以直接通过计算向量间距离来计算词语间的相似度。然而，该方法通常需要较大的数据集来进行训练，且对所有词语进行向量化，时间复杂度较高。而于本文的属性树构建而言，只需对数量不多的属性词进行向量化即可，因此本文采用文献[3]中的语境向量（context vector）作为属性词的向量表示，既能保留属性之间的关联，又能提高时间效率。具体的，语境向量计算步骤如下：

1. 将评论语料按句划分，假设共有 M 条评论语句。评论语句的集合记为 $S = \{S_1, \dots, S_M\}$ ，从这些语句中提取出的 N 个商品属性集合记为 $F = \{F_1, \dots, F_N\}$
2. 对每条语句规定一个 N 维特征向量，若第 i 个属性词出现在该语句中，则向量的第 i 维为 1，否则为 0。记所有语句的特征向量集合为 V ，对于 S_m 的特征向量 \vec{v}_m

$$(\vec{v}_m)_j = \begin{cases} 1 & S_m \text{ contains } F_j \\ 0 & \text{else} \end{cases} \quad (6-1)$$

3. 对每个属性词 F_n ，其语境向量如下计算：将所有包含 F_n 的句子的特征向量相加，并将结果第 n 维置为 0，即 F_n 的语境向量

$$\vec{w}_n = D \sum_{v \in V, v_n=1} \vec{v} \quad (6-2)$$

其中 D 是对角阵，除 D_{nn} 为 0 外对角线上每个元素均为 1，即

$$D = (d_{ij})_{N \times N} = \begin{cases} 1 & i = j \neq n \\ 0 & \text{else} \end{cases} \quad (6-3)$$

语境向量表征了属性之间的共现频率关系，第 n 个属性词 F_n 的第 m 维表征了第 n 个属性词与第 m 个属性词在同一句子中共现的频率。语境向量的这一特点保留了属性之间的关联性，在一个属性词的语境向量中，与该属性词关联性强的其他属性词对应维度的数值较大，而与之关联性弱的其他属性词对应维度的数值较小。因此用语境向量进行 hierarchical clustering 可以得到较为可靠的层次结果。

6.3 向量预处理

为避免语境向量中某些极端值对聚类效果的影响（例如，若某个特征语境向量某一维度过大，导致在计算距离时其它维度的作用不明显，不利于分类），因

此对语境向量进行对数预处理，具体地，对每个属性词的语境向量 F_n 作如下处理

$$(F_n)_j' = \begin{cases} \ln((F_n)_j + 1) & (F_n)_j > 0 \\ 0 & (F_n)_j \leq 0 \end{cases} \quad (6-4)$$

同时，商品名称作为出现频率最高的属性词，导致每个属性词与商品名称共现的频率远远高于与其他属性共现的频率，因此这一维度对于分类问题而言没有意义，反而由于其数值较大影响分类效果，因此将语境向量中与商品名称有关的维度去掉。

这样的预处理可以平滑数据较大的维度，减小极端数据对全剧的影响，使得分类结果更为准确。

6.4 自适应二分聚类

属性词的语境向量预处理完成后，对它们进行自顶向下的二分聚类形成一棵初级的二叉属性树。本文使用 UMN 大学的 George Karypis 等人提供的 CLUTO 聚类工具箱（见文献[11]）进行聚类操作。该工具箱提供了多种经典聚类算法的实现。然而，这些算法都需要预先设定分类类别数 k ，灵活性不好。为了实现自适应的分类，根据文献[3]中的方法，采用类内相似度阈值来控制分类，并自顶向下二分实现聚类。类内相似度的定义如下：

对于一个包含 n 个向量的集合 $C = \{\vec{v} | \vec{v} \in R^N\}$ ，其类内相似度

$$IS = \frac{\sum_{\vec{u}, \vec{v} \in C, \vec{u} \neq \vec{v}} \text{sim}(\vec{u}, \vec{v})}{n(n-1)} \quad (6-5)$$

其中 $\text{sim}(\vec{u}, \vec{v})$ 定义了两个向量 \vec{u} 和 \vec{v} 的相似度，本文采用夹角余弦值，即

$$\text{sim}(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \|\vec{v}\|} \quad (6-6)$$

具体聚类方法如下：

令 C_1, C_2, \dots, C_k 分别表示聚类得到的 k 类属性, $C_1 \cup C_2 \cup \dots \cup C_k = F$ (F 为所有属性词的集合), 记 $C = \{C_1, C_2, \dots, C_k\}$ 是各属性类构成的集合。

0. 令 $C = \{C_1\} = \{F\}$
1. 计算 C 中各类别的类内相似度 IS
2. 设定阈值 th (实验中取 0.7), 记 C 中类内相似度 $IS < th$ 的类别集合为 C'
3. 对 C' 中的 m 类属性分别用 `cluto` 工具箱进行 2 分类, 得到新的 $2m$ 类属性, 记作集合 C''
4. 令 $C = C - C' + C''$
5. 重复 1-4 步骤, 直至集合 C 不再变化, 所得 $C = \{C_1, C_2, \dots, C_k\}$, 即为最终聚类结果。

这样的聚类方法具有很好的自适应性, 不需要预先设置分类类别数, 而是对于类内相似度不高的属性类不断二分, 直至每个属性类都具有较好的内部相似度。聚类过程是自顶向下不断二分, 因此得到的是一棵二叉树。

6.5 类别标签提取

上述方法得到的二叉属性树, 所有属性词均出现在叶子结点, 非叶子结点没有定义, 这些非叶子结点, 其实是下辖叶子结点共同的子孙结点, 这些非叶子结点应该是对下辖叶子结点的抽象, 或者说, 是其子孙结点所组成的一类属性的类别标签。对于类别标签的提取, Shi B 等人的工作采用了 `overlap` 向量来提取类别标签 (见文献[3]), 具体如下:

1. 计算该类别所有属性词的共现矩阵 $A \in R^{N \times N}$ (N 为该类别的属性词数量), 即 A_{ij} 是该类别第 i 个属性词与第 j 个属性词在同一句子中共同出现的频次。矩阵 A 的第 i 行是表现第 i 个属性词与其他属性词的共现特征的向量。
2. 计算 `overlap` 向量 \vec{o} , \vec{o} 是一个二值向量, 若矩阵 A 的第 j 列中, 有超过 $p \times N$ 个值大于 0, 则 $\vec{o}_j = 1$, 否则 $\vec{o}_j = 0$ 。即

$$\vec{o}_j = \begin{cases} 1 & \frac{\# \{i | A_{ij} > 0\}}{N} > p \\ 0 & \text{else} \end{cases} \quad (6-7)$$

实验中取 $p = 0.5$, 即若该类属性词中, 第 j 个属性词与 50%以上的同类属性

词均在同一句子中出现过，则在 **overlap** 向量中对应维度 \vec{o}_j 为 1，否则为 0。

3. 以 **overlap** 向量作为掩膜，对矩阵 **A** 每一行做加权，得到矩阵 **A'**，即

$$A'_{ij} = A_{ij} \times \vec{o}_j \quad (6-8)$$

4. **A'** 中含有非零值个数最多的一行对应的属性词即该类属性词的类别标签。即

$$k = \max_i (\# \{j | A'_{ij} > 0\}) \quad (6-9)$$

该类别中第 k 个属性词即为类别标签。

若有多个非零值个数最多的行，则取非零值之和最大的一行作为类别标签。即若上述计算中 k 有多个取值 $K = \{k_1, k_2, \dots, k_n\}$ ，则

$$k = \max_{k \in K} \sum_{j=1}^N A'_{kj} \quad (6-10)$$

该方法中，**overlap** 向量为 1 的维度所对应的属性词是与较多同类属性词在同一语句中出现的属性词，而与最多属性词共现的属性词就是最终的类别标签。

然而，经我的实验，这样的方法提取的类别标签很不精确，存在许多错误，准确率较低。究其原因，在于 **overlap** 向量本质上只利用了共现频率这一低层次信息，没有任何语法语义相关的信息，因此难以得到较高的准确率。基于这一点考虑，我利用第 5 章中提取出的粗糙的从属关系，提出了一种新的标签提取方法，能够达到较好的效果。具体方法如下：

记第 5 章中提取的“主-从”属性对集合为 $R = \{ \langle m, s \rangle | m \in F, s \in F, m \text{ 是 } s \text{ 的主属性} \}$ 。

1. 对某一属性类的属性词集合 C_k ，根据第 5 章中提取出的从属关系，将该类别所有属性词的所有主属性词加入标签词候选集合 $M = \{m | \exists s \in C_k, s.t. \langle m, s \rangle \in R\}$ 。

2. 对每一个候选词 $m \in M$ ，按如下方式计算其作为该类属性词标签的得分：

$$score(m) = \sum_{s \in C_k, \langle m, s \rangle \in R} freq(s) \times freq(\langle m, s \rangle) \quad (6-11)$$

其中， $freq(s)$ 表示属性词 s 在评论语料库中出现的频次， $freq(\langle m, s \rangle)$ 表示“ $m-s$ ”作为主从关系（详见第 5 章中的两条规则）的属性在评论语料库中出现的频次。

3. 将候选词按得分从高到低排序，最高得分的候选词作为该类属性的标签。

实验表明，这样的标签词提取方法，由于应用了语义层面的高层次信息，使得标签词更加准确，详见实验结果分析。虽然在提取属性之间从属关系时，只应用了两条简单的语法依赖关系，得到的从属关系较为粗糙，存在一些错误，但是由于在提取标签词时利用的是其经统计处理后的信息，因此总体上较为准确。

6.6 小结

至此，我们采用合理的方式对属性词进行向量化，进行了适当的预处理，采用自顶向下自适应二分聚类的方法生成了一棵二叉树，并创造性地基于语法规则抽取属性间从属关系，并利用从属关系的统计特性对属性树中没有定义名称的非叶子结点进行了类别标签提取，得到了一棵初始的二叉属性树。

然而，由于二分方法的局限，得到的属性树恒为二叉树，这样的树形结构不够合理，许多结点所处的层次太深。考虑到错误提取的非属性词和聚类时被错误分类的属性词，初始的属性树准确性也有待提高。

第7章 最终属性树构建

7.1 初始属性树中存在的问题

经前述算法步骤得到的二叉属性树存在诸多问题，其中，最为明显的是两个问题：第一个问题是树形结构不合理，自适应二分只能得到二叉树，然而商品的属性层次结构一般来说不是二叉树形的，例如，对智能手机而言，其根结点为“phone”，按上述方法得到的初始属性树，第二层结点只有两个属性，

“screen”和“price”，而“battery”、“camera”、“quality”等和它们属于同一层次的属性，由于树形结构不合理，只能出现在更深层次的结点上，这导致了許多本不具有“主-从”关系的属性成为父子结点，如“screen-screen”、“screen-camera”等。第二个问题是，一些叶子结点的存在不合理，有些是错误归类的属性词，例如“battery”被归类在“screen”类别下，有些是非商品属性词，被错误地抽取出来并进行分类，且没有被滤除，例如“time”、“user”等词。

7.2 属性树的校正、完善与修剪

针对上述问题，本文提出如下 5 条规则，用以对初始的属性树进行校正、完善和修剪，具体如下：

1. 所有孩子结点均非叶子结点的结点，应被其孩子们取代

如若某个结点的孩子结点均非叶子结点，说明它的孩子们本就各自是不同的属性类，只是受到二分方法的限制被聚到一起，因此它们的共同父结点没有太大意义，是一个多余的中间层，应当被略去。例如图 2.2 所示例 1-1。

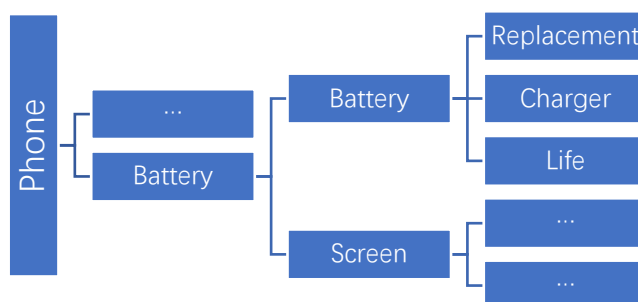


图 2.2 例 1-1

显然第三层的“battery”和“screen”各自是一类属性，由于二分方法，在第一次二分时将它们分开，使得两者被迫成为第三层的结点，且有一个多余的“battery”父结点。采用该规则后，树形结构变化如图 2.3 例 1-2。

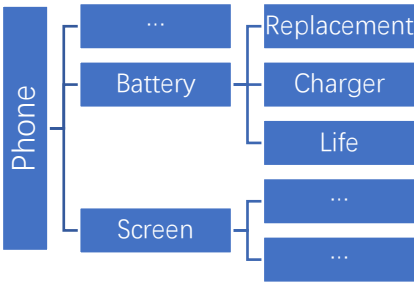


图 2.3 例 1-2

“battery”和“screen”成为“phone”的孩子结点，使得树形结构更加合理。

2. 在非叶子结点出现的属性词，应从叶子结点中删除

在生成初始属性树时，所有属性词均会成为叶子结点，在提取类别标签后，有些属性词会成为非叶子结点，说明这样的属性词本就是一个较高层次的属性，它的下层还有更细致的特征，因此不宜作为叶子结点重复出现，例如智能手机中的“battery”、“screen”、“camera”等等，这类属性词应从叶子结点中删除，这样能剔除属性树中一些不合理的上下层关系。例如图 2.4 所示例 2-1。

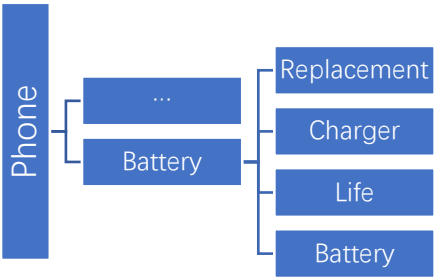


图 2.4 例 2-1

在这种情况下，“battery”、“life”、“charger”、“replacement”被归类到同一属性类，并且提取的标签词（父结点）是“battery”，这就导致了“battery-battery”这样的不合理上下层关系出现。“battery”这一属性词在非叶子结点出

现，因此它是一个较高层次的属性，不应再出现在叶子结点。应用该条规则，能有效滤除这样的不合理上下层关系，对于上述例子，应用规则后，变为图 2.5 所示例 2-2。

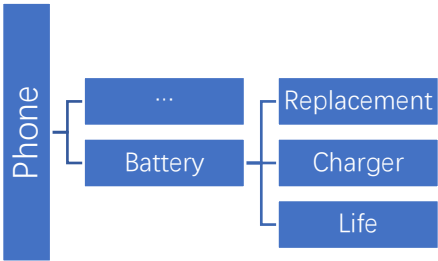


图 2.5 例 2-2

将高层次的属性从低层次的叶子结点中删除，能减少属性树中错误的上下层关系，使得属性树结构更加合理。

3. 删除与兄弟结点及父结点之间均无从属关系的叶子结点

属性树表现的是商品的属性结构，从高层到低层是属性逐渐细化的过程，因此在属性树中，父结点应是子结点的主属性，子结点应是父结点的从属性。利用这一点，可以将一些不合理的叶子结点属性词滤除。严格来说，不是父结点的从属性的叶子结点都该滤除，但是考虑到抽取的从属关系较为粗略，有一定的遗漏和错误，因此将条件从“非父结点的从属性的叶子结点应删除”放松到“与所有兄弟结点及父结点均无从属关系的叶子结点应删除”。

该规则对于滤除错误提取的属性词有较大帮助。使用 **double propagation** 抽取商品属性时，形如“**time**”这样的词，本不是商品属性词，但常被“**good**”一类的情感词修饰（“**good time**”），且出现频率较高，被错误地当做商品属性提取出来，并成为属性树的叶子结点。然而由于“**time**”这样的词与其他属性之间不具备从属关系，因此使用该规则可以轻松滤除。另外，该规则对于那些被分错类别的属性词也有滤除作用，因为分错类别的属性词与所在类别的其他属性词之间关联不大。例如图 2.6 所示例 3-1。

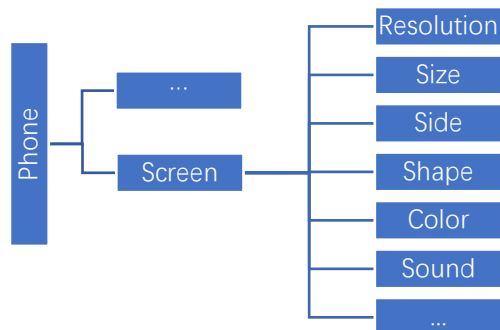


图 2.6 例 3-1

在这个例子中，“screen”这一属性类下，有诸多从属性，其中“resolution”、“size”、“shape”、“color”等都是正确的从属性，而“side”一词，则是由于评论预料中存在的诸如“on the positive/negative side”这样的短语，“side”被带有情感色彩的形容词修饰，被错误当做属性词提取出来，并进行了分类，从而出现在该类别下；而“sound”一词，则是被错误分类的属性，在抽取出的从属关系中，它的主属性应该是“phone”或“camera”，与“screen”并无关系。因此，应用该规则后，这两个错误的叶子结点能被滤除，如图 2.7 所示例 3-2。

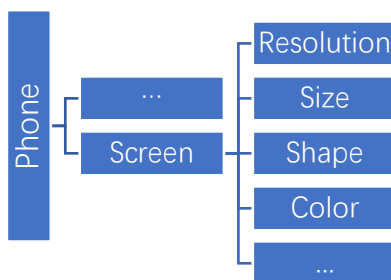


图 2.7 例 3-2

4. 仅有一个孩子结点的结点，应被其孩子结点取代

应用前述几条规则后，属性树中可能出现一些只有一个孩子结点的结点，这些结点属于多余的中间层，应被其孩子结点取代，例如图 2.8 所示例 4-1。

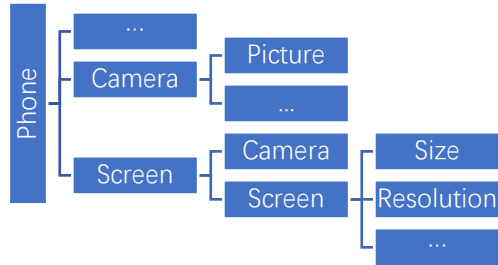


图 2.8 例 4-1

应用规则 2 后，将“Screen”属性类别下错误出现的“Camera”滤除后，出现了下面图 2.9 例 4-2 所示的结构：

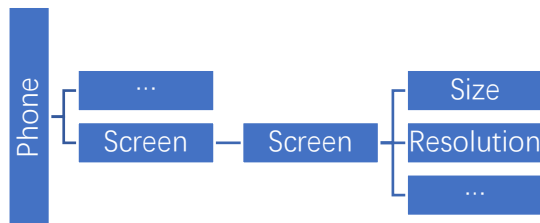


图 2.9 例 4-2

可以看出，第二层的结点“Screen”只有一个孩子结点“Screen”，可见是多余的，应用该规则后，可以剔除这样的结点，如图 2.10 所示例 4-3。

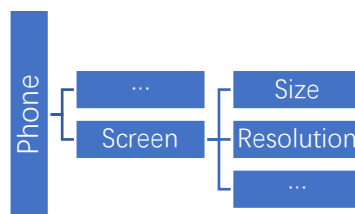


图 2.10 例 4-3

5. 同名的非叶子兄弟结点合并

在自适应二分类生成初始属性树时，通过不断二分类使得每一类的类内相似

度高于阈值，然而，这可能导致过度划分的情况，即，同一类属性被错误地划分为两类甚至多类。对于在这样的情况，由于被过度划分的两类或多类属性本质上是同一类属性，因此它们的标签词（父结点）应该是一致的，可以通过合并同名的兄弟结点将它们重新聚合，如图 2.11 所示例 5-1。

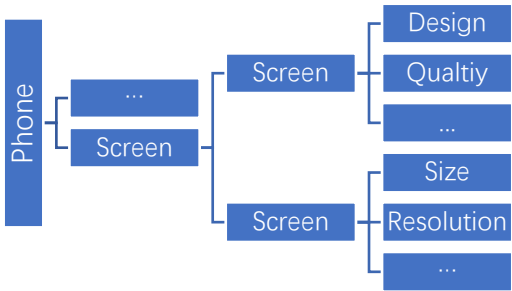


图 2.11 例 5-1

在这一例子中，本来第二层的“Screen”已经是不该再分的属性类，然而由于类内属性总体相似度未达到设定阈值，导致其继续划分成了两类属性，但由于两类属性本质上是同一类，两类的类别标签有较大可能一致，本类中，两类属性词标签都是“screen”，因此可以基于该条规则对其合并，得到图 2.12 所示例 5-2。

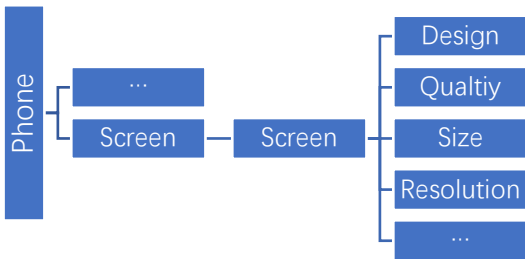


图 2.12 例 5-2

再应用规则 4 将多余的中间层“screen”去掉，即可得到正确的结果，如图 2.13 所示例 5-3。

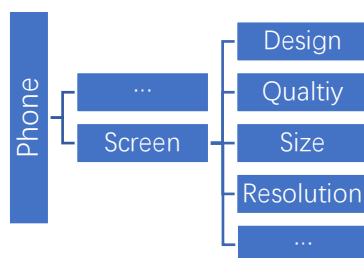


图 2.13 例 5-3

7.3 属性树各结点权重计算

经上述步骤，我们得到最终的属性树。属性树有许多应用场景，最常见的，是与情感二元关系分析相结合，分析评论语句对于商品各个属性评价的情感极性。然而，如果我们希望通过某一条具体的评论来推断该评论对应的人对于该商品各个属性的看法时，会遇到一个问题：一条评论很难涵盖属性树中商品的各方面属性，我们如何判断作者对那些没有提及的属性的态度呢？

对于这个问题，需要应用情感推断，即从已有的情感极性推断未知的情感极性。具体的，最简单的推断有如下两条规则：

1. 若评论文本对属性树某一结点有正向或负向的情感倾向，那么可以推断，该结点的子结点当中，评论文本没有表现出情感倾向的结点，其情感倾向与父节点一致。例如，若某评论语句提到了“the screen is poor.”，可知作者对“screen”的评价是负面的，若作者没有提到与“display”相关的评价，则“display”在属性树中作为“screen”的子结点，可以推断作者对其的评价也是负向的。
2. 若评论文本对属性树某一结点的各个子结点没有相反的情感倾向，那么可以推断，作者对该结点属性的情感倾向与其带有情感倾向的子结点一致。例如，若某评论语句提到“the screen is big and clear, and the camera works well.”，可知作者对“phone”的子结点“screen”和“camera”的评价均为正向，且评论文本没有提及其他“phone”的子结点，则可推断作者对“phone”的整体评价是正向的。

这两条规则可以进行一些简单的情感推断。然而对于较为复杂的情况，例如某评论文本对“screen”的评价是正向的，对“camera”和“battery”的评价是

负向的，此时该如何确定作者对“phone”这个整体的评价是正向还是负向？对此，为了方便情感计算及情感推断，我们为每个结点的属性赋予一个权重，代表其重要程度，当需要根据其子结点的情感倾向推断父节点的情感倾向时，可以通过对子结点情感倾向加权求和，作为父节点的情感倾向。

而对于结点重要程度的衡量，我们认为，被人们提及越多的属性，其重要程度越高。因此，我们的结点权重赋予规则确保每个结点的权重与其在所有评论文本中出现的频率成正相关关系。

具体地，对于一个结点 F_k ，设和它具有共同父结点的兄弟结点集合为 B 。则结点 F_k 对应的权重 $w(F_k)$ 为：

$$w(F_k) = \frac{freq(F_k)}{freq(F_k) + \sum_{f \in B} freq(f)} \quad (7-1)$$

其中函数 $freq(f)$ 是特征 f 在评论语料库中出现的频率。

基于这样的考量，我们可以对任意一个未知情感倾向的结点属性词进行推断。

具体地，对于没有情感倾向的结点 F_k ，设其具有正向情感倾向的子结点集合为 C_p ，具有负向情感倾向的子结点集合为 C_n ，则如下推断 F_k 的情感倾向 o （ o 为1代表正向， o 为-1代表负向， o 为0代表中性）：

$$o = Sgn(\sum_{f \in C_p} w(f) - \sum_{f \in C_n} w(f)) \quad (7-2)$$

其中 $Sgn(x)$ 是符号函数， x 为正取1， x 为负取-1， x 为0取0。

根据这样的推断方法，可以很好地辅助对于评论文本的情感分析，推断隐藏情感倾向。

7.4 小结

本章我们对树形结构不合理、准确度不佳的初始属性树进行了校正、完善和修剪，具体通过5条自定义规则来实现。这些规则中第1、4、5条规则用于改善树形结构，将原来死板的二叉树结构合理化；第2、3条规则用于剔除不可靠的结点，提升属性树的整体准确率。

经上述5条规则的反复应用，我们对属性树进行了校正、完善和修剪，使其树形结构更加合理，同时这些规则能滤除一些不合理的结点，使得属性树精

准性有所提高。

此外，我们还基于属性词在语料库中的统计特征为其赋予相应的权重，以代表属性词的重要程度，借此可以对评论文本中未明确表达情感倾向的属性词推断作者的情感倾向，获得更丰富的信息。这对于情感分析有很大帮助。

第8章 实验结果及评价

8.1 实验数据集介绍

本文实验所使用的数据，是从美国亚马逊网站上爬取的商品评论自由文本，主要分为智能手机和笔记本电脑两个领域。数据量大小见表 3.1。

表 3.1 数据量大小统计

数据集	商品数量	总评论数	总句子数	总单词数
智能手机	40	39672	386837	5072488
笔记本电脑	49	33264	189835	2522723

为了更清楚看到每条评论的平均情况，统计平均每条评论包含的句子数和平均每条评论包含的单词数，见表 3.2。

表 3.2 评论语句平均情况统计

数据集	平均每条评论句子数	平均每条评论单词数
智能手机	9.8	13.1
笔记本电脑	5.7	13.3

可以看出，相比于智能手机数据集，笔记本电脑数据集虽然语句长度与之类似，但平均每条评论所包含的语句较少。说明笔记本电脑数据集的评论大都较为简短，涉及的方面较少，因此提及的属性大都较为粗略，这也导致实验中发现笔记本电脑数据集的实验结果相比智能手机数据集表现差一些。

8.2 属性树结果展示

以下展示以智能手机数据集得出的结果为例。

以 Shi B 的算法，在本文智能手机数据集上对提取出的属性词转化的语境向量不断二分聚类，并用 overlap 向量提取标签，得到的智能手机属性树如图 3.1 所示。

可以看出，该方法得出的树形结构很不合理，二分导致树的深度过深而宽度

不足，且整体准确度较低，标签词也大都不准确。

而根据本文算法，经第 4 章~第 6 章的属性词抽取和向量化自适应二分聚类过程，再经基于从属关系的属性词标签提取，得到的初始属性树如图 3.2 所示。

因为都是对语境向量二分聚类的结果，所以本文的初始属性树树形结构与 Shi B 的结果完全一致，但标签词语是利用从属关系提取的，这一点与其采用的 overlap 向量不同，可以看到，标签词语的准确性相比于前述工作已有较大提高（具体分析见 3.3.3 节），再经第 7 章中所述规则对属性树进行校正、完善、修剪，最终生成的智能手机的属性树如图 3.3 所示。



图 3.1 Shi B 的属性树



图 3.2 本文初始属性树



图 3.3 本文最终属性树

可以看出，经校正、完善、修剪后的属性树，相较于 Shi B 的属性树和本文的初始属性树，树形结构更加清晰合理。属性树的宽度有所增加，丰富了每个层次的属性，且属性树的深度有很大削减，消除了许多多余的中间连接层。且由于使用了多种滤除机制，删掉了一些不合理的结点，属性树整体准确性也大有提高。

8.3 属性树结果分析

8.3.1 评价指标

由于评价体系是树状的，对于其准确度的衡量不便直接考虑，因此从属性树的各个父子结点之间的关系是否符合上下级关系来评价属性树的准确性。具体地，设属性树中共有 N 对父子结点，其中父结点在语义上是子结点的主属性的有 n 对，可以以正确父子结点对所占的比例 n/N 来衡量属性树的整体正确率。

8.3.2 整体结果分析

在智能手机和笔记本电脑两个数据集上得到的 Shi B 的属性树、本文初始属性树、本文最终属性树三者之间整体正确率对比见表 3.3（智能手机数据集）和表 3.4（笔记本电脑数据集）。

表 3.3 智能手机数据集结果

属性树	父子结点对总数	正确父子结点对数	总体正确率
Shi B 的属性树	66	28	42.4%
本文初始属性树	66	41	62.1%
本文最终属性树	34	27	79.4%

表 3.4 笔记本电脑数据集结果

属性树	父子结点对总数	正确父子结点对数	总体正确率
Shi B 的属性树	104	32	30.8%
本文初始属性树	104	46	43.8%
本文最终属性树	36	24	66.7%

从实验结果可以看出，不论在智能手机数据集或是笔记本电脑数据集上，本文的算法对于属性树的准确性都高于 Shi B 的方法生成的属性树，具体地，3.3.3 和 3.3.4 从初始属性树和最终属性树两个方面对比分析准确性提高的原因。3.3.3 节对比了初始属性树与 Shi B 的属性树的差异并分析了属性从属关系在其中起到的作用，3.3.4 节对比了初始属性树与最终属性树的差异，说明了本文对属性树进行校正的若干规则的作用。

8.3.3 初始属性树对比Shi B的属性树

初始属性树采用了 Shi B 的方法，对属性词提取语境向量，并二分聚类得到树形结构，区别在于，Shi B 的方法对于非叶子结点提取标签词采用 **overlap** 向量，而本文基于从属关系的统计特征提取。**overlap** 向量本质上是对频率共现信息的利用，而基于从属关系的统计特征，利用了语义层面的信息，因此可以达到较好的效果。实验结果验证了从属关系对于标签词的准确度大有帮助。具体地，在智能手机数据集和笔记本电脑数据集上分别对属性树中本文方法中得到的标签词与 Shi B 的 **overlap** 向量方法得到的标签词的准确度单独做分析，得到表 3.5 和表 3.6。

表 3.5 智能手机数据集上标签词正确率对比

属性树	标签词总数	正确标签词数	标签词正确率
Shi B 的属性树	19	5	26.3%
本文初始属性树	19	8	42.1%

表 3.6 笔记本电脑数据集上标签词正确率对比

属性树	标签词总数	正确标签词数	标签词正确率
Shi B 的属性树	53	16	30.2%
本文初始属性树	53	23	43.4%

可以看出，初始的属性树相较 Shi B 的结果，在对标签词的提取上，准确性已经有所提高。这是因为初始的属性树在标签词的提取上利用了从属关系这一信息，所以，虽然第 5 章中提取的从属关系较为粗略，但实验证明利用其统计信息对属性树标签词的提取是可靠的。

8.3.4 最终属性树对比初始属性树

虽然本文的初始属性树相较 Shi B 的属性树准确性上已经有所提高，然而不论 Bin Shi 的属性树，还是本文的初始属性树，都有一些显著的缺陷。

从树形结构上看，树的深度太深导致不必要的标签词语过多，例如智能手机的初始属性树中，“battery”、“camera”、“screen”等本该直接作为根结点“phone”的子结点，实际上却由于二分的算法导致它们处在第三、四甚至更深层的结点上，中间有许多多余的层次。这不仅导致属性树的整体准确性降低，也带来了结点数量过多的问题。例如在上述实验中，是对抽取出的属性词中频率最高的 50 个词进行聚类生成属性树，且这 50 个词中还含有少部分被错误抽取的非属性词，结果生成的初始树却包含超过 100 个结点，其中多余的非叶子结点占了很大比例。

另外，从准确性来看，初始属性树中存在较多错误的叶子结点（可能是非属性词，可能是错误归类的属性词），这也带来准确度低的问题。

而通过自定的 5 条规则对属性树进行校正、完善和修剪后，较大程度解决了上述问题，最终的属性树中，非叶子结点数量大大减少，且非叶子结点的标签词的准确度和整棵属性树的准确度都大大提高，具体见表 3.7（智能手机数据集）和表 3.8（笔记本电脑数据集）。

表 3.7 智能手机数据集上初始树和最终树对比

属性树	非叶子结点数	标签词正确率	属性树整体正确率
本文初始属性树	19	42.1%	62.1%
本文最终属性树	8	87.5%	79.4%

表 3.8 笔记本电脑数据集上初始树和最终树对比

属性树	非叶子结点数	标签词正确率	属性树整体正确率
本文初始属性树	53	43.4%	43.8%
本文最终属性树	11	72.7%	66.7%

可以看出，相比于初始属性树，最终属性树的表现有较大提升，多余的非叶子结点被滤除，标签词正确率和属性树整体正确率都大大提升。可见基于本文提出的规则对属性树进行的校正、完善、修剪是有效的。

8.4 小结

我们从两个方面对结果进行了分析。

一方面是本文的初始属性树与 Shi B 的属性树对比，两棵属性树的树形结构相同，均是由属性词的语境向量自顶向下自适应二分得到的层次结构，但本文提出了基于属性间从属关系提取标签词的方法，来替代传统的 overlap 向量提取标签词的方法。相较而言，overlap 向量的方法仅仅利用了低层次的共现频率信息，而本文的方法利用了从属关系这样语义层面的高层次的信息，因此得到的标签词更符合人们的认知。实验结果也验证了该方法的有效性。

另一方面是本文的最终属性树与本文的初始属性树对比。在第七章中，本文提出若干精细化的规则对初始属性树进行处理，重整了属性树的结构，滤除了许多多余的非叶子结点和错误的叶子结点，得到的最终的属性树在树形结构和准确度上都有较大提升。经实验结果验证，最终的属性树相比于初始属性树，标签结点的准确率和父子结点对的正确率都有较大提升，可见基于本文提出的规则对属性树进行的校正、完善、修剪是有效的。

第9章 总结与展望

本文在前人工作基础上，提出一种基于自由评论文本的、领域通用的、准确的层次结构评价体系构建方法。具体分为属性抽取、属性词向量化、向量聚类生成初始属性树、自定义规则校正修剪属性树等若干步骤。

本文的创新点在于，提出抽取属性间的从属关系，并利用它来修正初始属性树的标签，并提出若干自定义规则校正、完善和修剪属性树，使其层次结构更加合理，并提高其总体精确度。

今后，可将该层次结构评价体系的构建与文本的情感计算相结合，能够应用于数据挖掘、情感分析、情感推断等诸多领域，具有较高的实用价值。

插图索引

图 2.1	“double propagation”核心步骤图解.....	7
图 2.2	例 1-1	16
图 2.3	例 1-2	17
图 2.4	例 2-1	17
图 2.5	例 2-2	18
图 2.6	例 3-1	19
图 2.7	例 3-2	19
图 2.8	例 4-1	20
图 2.9	例 4-2	20
图 2.10	例 4-3	20
图 2.11	例 5-1	21
图 2.12	例 5-2	21
图 2.13	例 5-3	22
图 3.1	Shi B 的属性树	27
图 3.2	本文初始属性树	28
图 3.3	本文最终属性树	29

表格索引

表 3.1 数据量大小统计	25
表 3.2 评论语句平均情况统计	25
表 3.3 智能手机数据集结果	30
表 3.4 笔记本电脑数据集结果	30
表 3.5 智能手机数据集上标签词正确率对比	31
表 3.6 笔记本电脑数据集上标签词正确率对比	31
表 3.7 智能手机数据集上初始树和最终树对比	32
表 3.8 笔记本电脑数据集上初始树和最终树对比	32

参考文献

- [1] Hu, M., & Liu, B. (2004). Mining opinion features in customer reviews. National Conference on Artificial Intelligence (Vol.69, pp.755-760). AAAI Press.
- [2] Yu, J., Zha, Z. J., Wang, M., Wang, K., & Chua, T. S. (2011). Domain-Assisted Product Aspect Hierarchy Generation: Towards Hierarchical Organization of Unstructured Consumer Reviews. Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, Uk, A Meeting of Sigdat, A Special Interest Group of the ACL (pp.140-150). DBLP.
- [3] Shi, B., & Chang, K. (2008). Generating a concept hierarchy for sentiment analysis. 312-317.
- [4] Yu, J., Zha, Z. J., Wang, M., & Chua, T. S. (2011). Hierarchical organization of unstructured consumer reviews. International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April (pp.171-172). DBLP.
- [5] Ye, S., & Chua, T. S. (2006). Learning Object Models from Semistructured Web Documents. IEEE Educational Activities Department.
- [6] Qiu, G., Liu, B., Bu, J., & Chen, C. (2011). Opinion word expansion and target extraction through double propagation. Computational Linguistics, 37(1), 9-27.
- [7] Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. Meeting on Association for Computational Linguistics (pp.423-430). Association for Computational Linguistics.
- [8] Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, Usa, August (pp.168-177). DBLP.
- [9] Rohlf, F. J. (1970). Adaptive hierarchical clustering schemes. Systematic Zoology, 19(1), 58-82.
- [10] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. Computer Science.
- [11] Karypis, G. (2002). Cluto-a clustering toolkit. CLUTO - A Clustering Toolkit, 4(2), 163-165.

致 谢

感谢邓北星老师和黄永峰老师对我的帮助，让我能顺利完成这项工作。同时，感谢吴方照学长在毕业设计过程中给予我的宝贵意见和悉心指导。最后，感谢刘俊鑫学长提供的亚马逊爬虫软件，让我能轻松获取大量有用数据。

谢谢所有在毕业设计阶段给予我无私帮助的人！

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：_____ 日 期：_____

附录A 书面翻译

挖掘客户评论中的情感相关特征

摘要

在网上销售商品的商家请求客户对销售的商品和相关服务进行评论是一种常见的做法。随着电商受欢迎程度的增加，客户评论数量也随之增长。对于受欢迎的商品，评论数量可以达到数百条甚至更多。对于可能购买该商品的消费者，想要通过阅读所有评价来决定是否购买商品变得很困难。在这篇文章中，我们的目标是总结商品的所有客户评论。这个总结任务不同于传统的文本摘要，因为我们只对客户评论中带有情感倾向的商品特征感兴趣，以及对这些商品特征客户的评价是正面的还是负面的。与传统的文本摘要不同，我们不从评论中选取或重写原始句子的一部分来总结评论，以此捕捉他们的主要观点。在本文中，我们注重于挖掘客户评论中包含的情感相关特征。本文提出了一些方法来挖掘这些特征。我们的实验结果表明，这些方法是非常有效的。

引言

随着电子商务的迅猛发展，各类网站上开始售卖越来越多的商品，越来越多的人开始在网上购物。为了提高客户满意度和购物体验，卖家通常允许客户对他们购买的商品进行评价。随着越来越多的普通用户对因特网的熟悉，更多的人开始撰写评论。结果，商品收到的评论数量迅速增长。一些受欢迎的商品可以在一些大型购物网站上收到成百上千条评论。这使得对于那些可能购买这些商品的潜在客户，很难逐一阅读这些评论以便做出是否购买的决定。

在本文的研究中，我们提出研究基于商品特征的在线商品评论的观点摘要。该任务分两步：

1. 识别出那些客户对其表达出情感的商品特征（称作情感相关特征），并对这些特征根据其在评论中出现的频率排序。
2. 对每一个特征，我们识别出对其有正向情感和负向情感的评论数量。着有助于潜在客户浏览这些评论。

举一个简单例子来说明。假设我们总结了某一数码相机“数码相机_1”的评论，我们的摘要形式如下：

数码相机_1:

图片质量

正向: 253 <个人评论>

负向: 6 <个人评论>

尺寸

正向: 134 <个人评论>

负向： 10 <个人评论>

.....

“图片质量”和“尺寸”就是情感相关特征。共有 253 条客户评论对图片质量表达了正向情感，只有 6 条客户评论对其表达了负向情感。<个人评论>链接到对该特征表达正向（或负向）情感的特定评论。

有了这样一个基于特征的情感摘要，一个潜在客户可以轻松地看出已有客户对这个数码相机的感受。如果他/她对这个数码相机的图片质量很感兴趣，他/她还可以通过点击<个人评论>的链接深入查看已有客户为什么喜欢它或是为什么抱怨它。

我们的任务与传统的文本摘要（Radev and McKeown. 1998; Hovy and Lin 1997）在各个方面有着显著不同。首先，我们的摘要结构化的而不是像大多数摘要方法那样得到另一个更简短的自由文本。其次，我们只对顾客表达出情感倾向的商品特征及情感倾向的正负兴趣。我们不会像传统方法那样，通过选择或重写原评论中的部分语句来进行摘要，以此得出他们的主要观点。

在这篇文章中，我们只关注评论摘要任务的第一步。即，我们的目标是挖掘评论中提到的商品特征。任务的第二步，确定一个情感倾向是正向还是负向的将在随后的另一篇论文中讨论，因为这一步相当复杂。有人可能会问，“为什么不让卖家或者制造商直接提供一个商品特征的列表呢？”这是一种可能的解决方案。然而，这种方案有诸多问题：（1）一个卖家很可能同时销售许多商品，因此很难提供每个商品的特征列表。（2）对于同一个商品特征，卖家或厂商所用的词汇很可能与普通客户不同。这会导致在客户在寻找自己感兴趣的商品特征的信息时出现问题。此外，客户可能会对列表中缺乏某些特定的商品特征感到不满。（3）客户可能会对一些制造商根本不在意的特征进行评论，即某些预期之外的特征。（4）制造商可能不希望客户了解一些自己商品有缺陷的特征。

这篇文章提出许多基于数据挖掘和自然语言处理的方法来挖掘商品的情感相关特征。我们的实验结果表面这些方法是非常有效的。

相关工作

我们的工作主要与两个研究领域有关，文本摘要和术语识别。主流的文本摘要技术分为两类：模板实例化和文本抽取。前者的框架下已有工作包括（DeJong 1982），（Tait 1983）和（Radev and McKeown 1998）。他们专注于从文档中识别并抽取某些核心实体和因素，并将其包装在模板中。该框架需要背景分析来讲模板实例化到恰当的细节水平。因此，它不是领域通用的（Sparck-Jones 1993a, 1993b）。我们的方法不需要应用任何模板，而且是领域通用的。

文本抽取框架（Paice 1990; Kupiec, Pedersen 和 Chen 1995; Hovy and Lin 1997）识别出一些代表性的句子来总结全文。近年来，许多复杂的新方法被提出，例如，强话题概念（Hovy and Lin 1997），词汇链（Barzilay and Elhadad 1997）和话语结构（Marcu 1997）。我们的工作与之不同，因为我们不是提取最有代表性的句子，而只是识别并抽取那些具体的商品特征和与之相关的观点。

Kan 和 McKeown (1999) 提出了一种将模板实例化与句子抽取相结合的方法。

(Boguraev and Kennedy 1997) 也报告了一种在文档中搜寻几个非常突出的表达、对象或事件的方法，并借助它们来总结文档。我们的工作还是与之不同，因为我们需要搜寻一系列客户评论中包含的商品特征，不论它们是否突出。

大多数文本摘要的现有工作都是基于单篇文档。部分研究者也研究了多篇包含相似信息的文档的摘要。他们的主要目的是总结这些文档包含信息的异同 (Mani and Bloedorn 1997)。显然，我们的工作与之相关但不相同。

在术语识别的领域，有两种基本的方法用于发现语料库中的术语：依赖句法描述术语的符号方法，即名词短语，以及基于组成术语的名词之间彼此接近这一事实的统计方法 (Jacquemin and Bourigault 2001; Justeson and Katz 1995; Daille 1996; Church and Hanks 1990)。然而，使用名词短语往往会产生许多非术语，而使用重复出现的短语会漏掉许多低频的术语。我们基于关联规则挖掘的方法则不存在这些问题，而且因为我们只对客户表达情感的特征感兴趣，我们能够发现一些不常见的特征。

我们基于特征的情感摘要系统与这篇文章也有关联 (Dave, Lawrence and Pennock 2003)，在这篇文章中使用训练语料库构建了商品评论语句的语义分类器。然而，他们的系统并不挖掘商品特征。此外，我们的方法不需要使用训练语料库来构建摘要。

本文方法

Figure 1 给出了我们的情感摘要系统的架构概述。系统分两步进行摘要：特征提取和情感倾向识别。系统的输入是商品名称和商品的所有评论的入口页面。输出是引言部分所示的评论摘要。

给定输入，系统首先下载（或爬取）所有的评论，并将它们存入评论数据库。特征提取函数（本文的重点），先提取出那些许多人都在自己的评论中对其表达出观点的热门特征，再去寻找那些不常见的特征。情感倾向识别函数接收这些生成的特征，并将与特征相关的观点归纳为 2 类：正向和负向。在 Figure 1 中，POS tagging 是指自然语言处理的词性标注 (Manning and Schütze 1999)。接下来，我们轮流讨论特征提取中的每一个函数。

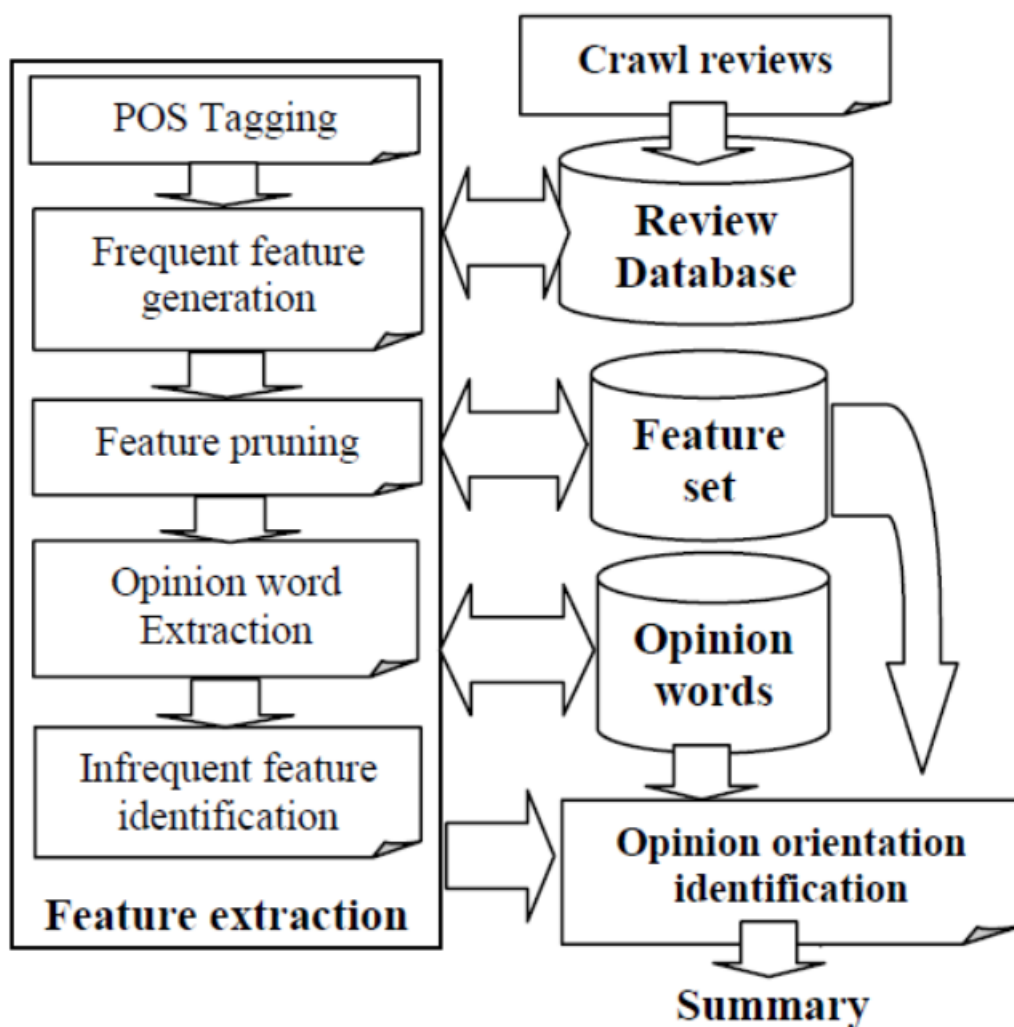


Figure 1: The opinion summarization system

词性标注 (POS)

在讨论自然语言处理中的词性标注的应用之前，我们先给出一些评论中的一些例句，以描述我们将要处理哪些观点。

我们的系统旨在找到人们喜欢给定商品的哪些特征和不喜欢给定商品的哪些特征。因此，如何找出人们谈论的商品特征是重要的一步。然而，由于自然语言理解上的困难，一些类型的语句很难处理。让我们分别看看数码相机评论中一些容易处理和较难处理的句子：

“The pictures are very clear.”

“Overall a fantastic very compact camera.”

在第一个句子中，客户对数码相机的图片质量感到满意，“picture”就是客户谈论的商品特征。相似的，第二个句子表面“camera”是客户表达观点的特征。尽管这两个句子中的商品特征是句子里面明确提到的，但也有一些句子中的特征是隐含的而不易发现。例如，

“While light, it will not easily fit in pockets.”

该客户在谈论相机的“size”（尺寸），但是单词“size”在句子中并没有明确提及。为了找到这样的隐含特征，需要语义理解，这就要求更多复杂的方法了。然而，隐含特征出现的频率远远低于显式特征。在这篇文章中，我们只关注寻找以名词或名词短语方式显式出现在评论中的特征。为了从评论中识别名词和名词短语，我们使用词性标注。

这篇文章中，我们使用 NLProcessor 语言解析器（NLProcessor 2000），它能解析每个句子并产生每个单词的词性标签（无论该单词是名次，动词，形容词，等等），并且它能识别的名词和动词组（句法分块）。以下是一个带有 POS 标签的句子。

```
<S> <NG><W C='PRP' L='SS' T='w' S='Y'> I </W></NG> <VG> <W C='VBP'> am
</W><W C='RB'> absolutely </W></VG> <W C='IN'> in </W> <NG><W C='NN'> awe </W>
</NG> <W C='IN'> of </W><NG> <W C='DT'> this </W> <W C='NN'> camera </W></NG><W
C='.'> . </W></S>
```

NLProcessor 系统生成 XML 格式的输出。例如< W C='NN'>表示一个名词，<NG>表示一个名词词组或名词短语。

每一个句子连同它每个单词的 POS 标签信息被一起保存在评论数据库中。

然后创建一个事物文件，用于在下一步中生成频繁出现的商品特征。在这个文件中，每行包含来自于一条语句中的单词，仅包含预处理后的该句的名词和名词短语。原因是句子的其他成分不太可能是商品特征。这里的预处理包括删除停用词，保留词干和模糊匹配。模糊匹配（Jokinen and Ukkonen 1991）用于处理单词变体和拼写错误。例如，

“autofocus”和“auto-focus”实际上是指相同的特征。所有出现的“autofocus”都被自动替换为“auto-focus”。

高频特征生成

这一步是为了找出人们最感兴趣的那些特征。为此，我们使用关联规则挖掘（Agrawal and Srikant 1994）来查找所有频繁出现的项目集。在我们的语境下，项目集是一组共同出现的单词或短语。

关联规则挖掘说明如下：

令 $I = \{i_1, \dots, i_n\}$ 为一些项目的集合， D 为一系列事件的集合（数据集）。每一个事件由 I 的子集构成。一个关联规则是一种蕴含形式 $X \rightarrow Y$ ，其中 $X \subset I, Y \subset I$ ，且 $X \cap Y = \emptyset$ 。我们称规则 $X \rightarrow Y$ 在 D 中以置信度 c 成立，当 D 中 $c\%$ 的事件既支持 X 也支持 Y 。我们称该规则在 D 中的支持度为 s ，当 D 中 $s\%$ 的时间包含 $X \cup Y$ 。关联挖掘规则的任务是，生成所有 D 中置信度和支持度高于人为设定的最小置信度和最小支持度的关联规则。

挖掘频繁出现的短语：将上述提取的每条信息存储在被称为事物集或事物文件的数据集中。然后，我们运行关联规则挖掘工具，CBA (Liu, Hsu and Ma 1998)，该工具是基于 Apriori 算法 (Agrawal and Srikant 1994) 实现。它查找事务文件中所有频繁出现的项目集。每个产生的项目集是一个可能的特征。在我们的工作中，我们将一个项目集定义为频繁的，当它出现在超过 1% (最小支持) 的评论语句中。

Apriori 算法分为两步。第一步，它从一组满足用户指定的最小支持率的事物集中查找所有频繁项目集。第二步，它从 的频繁项目集中生成规则。对于我们的任务，我们只需要第一步，即找到频繁项目集，这些将是候选特征。此外，我们只需从三个单词以内的项目集中查找频繁项目集，因为我们认为商品特征不会超过三个单词 (这个限制很容易放松)。

生成的频繁项目集 (本文中也成为候选频繁特征) 被存储到特征集中用于进一步处理。

特征滤除

并非所有关联挖掘产生的频繁特征都是有用的或是正确的。还有一些我们不感兴趣的冗余项。特征修剪旨在移除这些错误的特征。我们提出以下两种滤除方式。

紧致性滤除：该方法检查至少包含两个单词的特征，我们称之为特征短语，并从中删除无意义的特征。

在关联挖掘算法中，没有考虑一个项目 (单词) 在一个事件 (句子) 中出现的位置。然而，在自然语句中，共同出现且有明确顺序的单词更可能是有意义的短语。因此，关联挖掘生成的一些频繁特征短语可能不是真正的特征。紧致性滤除的思想是滤除那些所包含的单词不共同出现的候选特征短语。我们利用候选特征短语 (项目集) 中各个单词之间的距离来进行滤除。

定义 1：紧致短语

令 f 是一个频繁特征短语，且 f 包含 n 个单词。假设一个句子 s 包含 f ，且 f 中的单词在 s 中出现的顺序是： w_1, w_2, \dots, w_n 。如果在 s 中，以上顺序中任意两个相邻单词 (w_i 和 w_{i+1}) 的距离都不超过 3，则我们称 f 在 s 中是紧致的。

如果 f 在评论数据库中 m 个句子里出现，且它在 m 个句子中至少 2 个句子里是紧致的，则我们称 f 是一个紧致的特征短语。

例如，我们有一个频繁特征短语 “digital camera”，且有 3 个评论数据库中的句子包含该短语：

“I had searched for a digital camera for 3 months”

“This is the best digital camera on the market”

“The camera does not have a digital zoom”

短语 “digital camera” 在第一个句子和第二个句子中是紧致的，但在最后一个句子中不是。然而，它仍然是一个紧致短语因为它有两次在句子中以紧致的形式出现。

对于一个特征短语和一个包含该短语的句子，我们检查短语中每个单词在句子中出现

的位置，看它在该句子中是否是紧致的。如果在评论数据集中不能找到两个紧致的句子，我们就要滤除这个特征短语。

冗余滤除：在这一步，我们专注于滤除只包含一个单词的冗余特征。为了描述一个冗余特征，我们有如下定义：

定义 2： p-支持度（纯粹支持度）

特征 *ft* 的 p-支持度是满足这样条件的句子数量：*ft* 在这些句子中作为名词或名词短语出现，且这些句子中不能包含 *ft* 的超集。

p-支持度与关联挖掘中一般的支持度不同。例如，特征 “manual” 的支持度为 10 个句子。在评论数据集中它也是特征短语 “manual mode” 和 “manual setting” 的子集。假设这两个特征短语的支持度分别为 4 和 3，并假设这两个短语没有在任何一句子中同时出现，且所有的特征均是以名词或名词短语的形式出现。那么 “manual” 的 p-支持度就应该是 3。回想一下，我们要求特征均以名词或名词短语的形式，因为我们不想寻找形容词或副词作为特征。

我们运用最小 p-支持度来滤除那些冗余规则。如果一个单词的 p-支持度低于最小 p-支持度（在我们的系统中，设为 3）且该特征是另一特征短语的子集（这意味着该特征单独出现也许不是我们感兴趣的），它将被滤除。例如，“life” 本身不是一个有用的特征，然而 “battery life” 是一个有意义的特征短语。在前述例子中，“manual” 具有 3 的 p-支持度，它不会被滤除。这是一个合理的考虑，“manual” 有两层意思，作为一个名词表示 “references”，作为一个形容词表示 “of or relating to hands”。因此这三个特征，“manual”、“manual mode”、“manual setting”，都可能是感兴趣的。

情感词抽取

情感词是人们用来表达正向或负向情感的词语。观察到人们通常使用句子中位于特征词附近的情感词来表达他们对商品特征看法，我们可以使用所有修剪后剩余的频繁特征来从评论数据集中抽取情感词。例如，让我们看看下面两个句子：

“The strap is horrible and gets in the way of parts of the camera you need access to.”

“After nearly 800 pictures I have found that this camera takes incredible pictures.”

在第一个句子中，“strap” 这个特征靠近情感词 “horrible”。在第二个句子中，特征 “picture” 靠近情感词 “incredible”。

通过这些观察，我们采用以下方式抽取情感词：

对评论数据集中的每个句子，如果它包含任何频繁特征，则抽取其附近的形容词。如果找到这样的形容词，它被认为是一个情感词。附近的形容词是指相邻的形容词，它修饰作为频繁特征的名词或名词短语。

正如上述例子中，“horrible” 是修饰 “strap” 的形容词，而 “incredible” 是修饰 “picture” 的形容词。

我们用保留词干和模糊匹配来处理单词变种和拼写错误。这样，我们构建了一个情感

词列表，将在接下来的叙述中用到。

非频繁特征识别

对于给定商品，频繁特征是人们最感兴趣的热门特征。然而，也有一些特征只有少部分人提及。这些特征也可能有部分潜在客户会感兴趣。问题是，怎样抽取这些非频繁特征？考虑以下句子：

“Red eye is very easy to correct.”

“The camera comes with an excellent easy to install software.”

“The pictures are absolutely amazing”

“The software that comes with it is amazing”

语句 1 和语句 2 都包含相同的情感词 “easy”，尽管描述的是不同特征：语句 1 是关于 “red eye”，语句 2 是关于 “software”。假设在我们的评论数据集中 “software” 是一个频繁特征，“red eye” 是一个非频繁特征但是也是我们所感兴趣的。类似的，“amazing” 在语句 3 和 4 中都出现了，语句 3 是关于 “picture” 而语句 4 是关于 “software”。

从这个例子中，我们可以看出人们常用相同的形容词来描述不同的对象。因此，我们可以用情感词来寻找生成频繁特征的步骤中无法发现的特征。

在情感词生成的步骤中，我们用频繁特征来寻找邻近的修饰该特征的情感词。在这一步，我们用已知的情感词来搜寻附近的被情感词所修饰的特征。在这两个步骤中，我们都利用了观察到的“情感词倾向于紧挨着特征词共同出现”这一特点。我们用以下流程来抽取非频繁特征：

对评论数据集中的每条语句，如果它不包含频繁特征，但是包含一个甚至更多情感词，则找到该情感词最邻近的名词或名词短语。然后将该名词或名词短语作为一个非频繁特征存入特征集。

我们用最邻近的名词或名词短语作为被情感词修饰的名词或名词短语是因为大多数情况都是这样。由于查找情感词修饰的相应名词或名词短语需要自然语言理解，这仅仅使用 POS 标签是很难办到的，所以我们用这种简单的启发式方法来找到最邻近的名词或名词短语来替代。这很管用。

用情感词识别非频繁特征存在一个问题，就是它也可能识别出一些与所给商品不相关的名词或名词短语。原因是有些人会用一些常见形容词来描述许多对象，其中既有我们感兴趣的特征也有不相关的对象。考虑如下情况：

“The salesman was easy going and let me try all the models on display.”

“salesman”并不是商品的相关特征，但是它会因为附近的情感词 “easy” 而被当做一个非频繁特征。

然而，这并不是一个严重的问题，因为非频繁特征与频繁特征相比数量很小。它们约占我们实验结果中获得的特征总数的 15-20%。非频繁特征的生成是为了完整性。另外，频繁特征相比于非频繁特征更为重要。由于我们将特征按照它们的 p-支持度排序，那些错误

的非频繁特征将会被排到非常靠后的位置，因此不会影响到大多数客户。

句子情感极性识别：在情感相关特征被识别出来后，我们需要确定每个句子语义上的情感极性（正向或负向）。这包含两个步骤：（1）对情感词列表中的每个情感词，我们使用自助法和 WordNet（Miller et al. 1990）来识别它的情感极性，（2）然后我们基于句子中情感词的主导极性来确定每个句子的情感极性。细节将在随后的论文中介绍。

Table 1: Recall and precision at each step of the system

Product name	No. of manual	Frequent features		Compactness		P-support		Infrequent feature	
		Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Digital camera1	79	0.671	0.552	0.658	0.634	0.658	0.825	0.822	0.747
Digital camera2	96	0.594	0.594	0.594	0.679	0.594	0.781	0.792	0.710
Cellular phone	67	0.731	0.563	0.716	0.676	0.716	0.828	0.761	0.718
Mp3 player	57	0.652	0.573	0.652	0.683	0.652	0.754	0.818	0.692
DVD player	49	0.754	0.531	0.754	0.634	0.754	0.765	0.797	0.743
Average	69	0.68	0.56	0.67	0.66	0.67	0.79	0.80	0.72

实验

我们对五款电子产品的客户评论进行了实验：2 款数码相机，1 款 DVD 播放机，1 款 MP3 播放器，和 1 款手机。我们从 Amaon.com 和 C|net.com 这两个网站收集评论。这两个网站的商品有大量的评论。每条评优包含评论文本和标题。有一些额外的信息本项目中没有用到，包括日期、时间、作者姓名和地址（亚马逊的评论），还有评级。

对每个商品，我们先爬取并下载它的前 100 条评论，这些评论文档加以清理以去除 HTML 标签。之后，用 NLProcessor 来生成 POS 标签。应用我们的系统来进行特征抽取。

为了评估发现的特征，我们人工地阅读所有评论，并为每个商品制作了一个特征列表。这些特征在有情感倾向的语句中大部分都是明确的，例如 “the pictures are absolutely amazing” 中的 “pictures”。隐含特征例如 “it fits in a pocket nicely” 中的 “size” 人工提取也是非常容易的。Table 1 中的 “No. of manual features” 一行展示了每个商品的人工提取特征数量。

Table 1 给出了所有结果的准确率和召回率。我们评估了我们算法当中每一步的结果。在这张表中，第一列列出每一个商品。第 3、4 列给出了每个商品通过关联规则挖掘生成的频繁特征的召回率和正确率。结果表明频繁特征中包含许多错误，仅仅使用该步骤将导致一个不理想的结果，即较低的准确率。第 5、6 列展示了紧致性滤除应用后对应的结果。我们可以看出，通过该滤除规则后，准确率有显著提高，而召回率仍较为平稳。第 7、8 列给出使用 p-支持度滤除规则后的结果。准确率又有了戏剧性的提升。而召回率几乎不变。第 4-8 列的结果明显证明了这两个滤除方法的有效性。第 9、10 列给出了识别非频繁特征后的结果。召回率有了戏剧性的提高。准确率平均而言有一点点下降。然而，这并非一个主要问题，因为非频繁特征被排序得相当靠后，因此对大多数用户而言没有影响。

总之，从平均 80%的召回率和 72%的准确率来看，我们相信我们的方法是很有前途的，并可以在实际环境中使用。

结论

在这篇文章中，我们基于关联规则挖掘和自然语言处理方法提出了一系列方法来从商品评论中挖掘情感相关特征。目的是为在线销售的商品的大量客户评论提供一个基于特征的摘要。我们相信，随着越来越多的人在网上购买商品并表达他们的意见，这个问题会变得越来越重要。我们的实验结果表面，所提出的方法在执行这项任务上是有效的。在未来的工作中，我们计划进一步改进这些方法。我们还计划根据对特征表的情感强度来对特征分组，例如，用来确定客户强烈喜欢或不喜欢的特征。这将进一步改善特征提取和后续的情感摘要。

致谢

这项工作得到了国家自然科学基金 IIS-030739 的资助。

参考文献

- Agrawal, R. and Srikant, R. 1994. "Fast algorithm for mining association rules." *VLDB'94*, 1994.
- Barzilay, R., and Elhadad, M. 1997. Using lexical chains for text summarization. *ACL Workshop on Intelligent, scalable text summarization*.
- Boguraev, B., and Kennedy, C. 1997. Saliency-based content characterization of text documents. *In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*.
- Church, K. and Hanks, P. 1990. Word association norms, mutual information and lexicography. *Computational Linguistics*, 16(1) : 22-29.
- Daille. 1996. Study and Implementation of Combined Techniques for Automatic Extraction of Terminology. *The Balancing Act: Combining Symbolic and Statistical Approaches to Language Processing*. MIT Press, Cambridge.
- Dave, K., Lawrence, S., and Pennock, D., 2003. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. *WWW-2003*.
- DeJong, G. 1982. An Overview of the FRUMP System. *Strategies for Natural Language Parsing*. 149-176.
- Hovy, E., and Lin, C.Y. 1997. Automated Text Summarization in SUMMARIST. *ACL Workshop on Intelligent, calable Text Summarization*.
- Jacquemin, C., and Bourigault, D. 2001. Term extraction and automatic indexing. In R. Mitkov, editor, *Handbook of Computational Linguistics*. Oxford University Press.
- Jokinen P., and Ukkonen, E. 1991. Two algorithms for approximate string matching in static texts. In A. Tarlecki, (ed.), *Mathematical Foundations of Computer Science*.
- Justeson, J. S., and Katz, S.M. 1995. Technical Terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering* 1(1):9-27.
- Kan, M. and McKeown, K. 1999. Information Extraction and Summarization: Domain Independence through Focus Types. *Columbia University Technical Report CUCS-030-99*.
- Kupiec, J., Pedersen, J., and Chen, F. 1995. A Trainable Document Summarizer. *SIGIR-1995*.
- Liu, B., Hsu, W., Ma, Y. 1998. Integrating Classification and Association Rule Mining. *KDD-98*, 1998.
- Mani, I., and Bloedorn, E., 1997. Multi-document Summarization by Graph Search and Matching. *AAAI-97*.
- Manning, C. and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*, MIT Press. May

1999.

Marcu, D. 1997. From Discourse Structures to Text Summaries. *ACL Workshop on Intelligent, Scalable Text Summarization*.

Miller, G., Beckwith, R, Fellbaum, C., Gross, D., and Miller, K. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235-312.

NLProcessor – *Text Analysis Toolkit*. 2000. <http://www.infogistics.com/textanalysis.html>

Paice, C. D. 1990. Constructing Literature Abstracts by Computer: techniques and prospects. *Information Processing and Management* 26:171-186.

Radev, D. and McKeown, K. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469-500, September 1998.

Sparck-Jones, K. 1993a. Discourse Modeling for Automatic Text Summarizing. *Technical Report 290*, University of Cambridge.

Sparck-Jones, K. 1993b. What might be in a summary? *Information Retrieval* 93: 9-26.

Tait, J. 1983. *Automatic Summarizing of English Texts*. Ph.D. Dissertation, University of Cambridge.

原文索引

[1] Hu, M., & Liu, B. (2004). Mining opinion features in customer reviews. National Conference on Artificial Intelligence (Vol.69, pp.755-760). AAAI Press.

综合论文训练记录表

学生姓名		学号		班级	
论文题目					
主要内容以及进度安排					
	指导教师签字：_____ 考核组组长签字：_____ 年 月 日				

中期考核意见	<div>考核组组长签字：_____</div> <div>年 月 日</div>
指导教师评语	<div>指导教师签字：_____</div> <div>年 月 日</div>
评阅教师评语	<div>评阅教师签字：_____</div> <div>年 月 日</div>

答辩小组评语	<div style="text-align: right;">答辩小组组长签字：_____</div> <div style="text-align: right;">年 月 日</div>
--------	--

总成绩： _____

教学负责人签字： _____

年 月 日