**清 华 大 学**

**综 合 论 文 训 练**

**题目：社交网络中的谣言检测系统**

|  |  |
| --- | --- |
| 系 别： | 软件学院 |
| 专 业： | 计算机软件 |
| 姓 名： | 钟仰新 |
| 指导教师： | 刘世霞 副教授 |

2016年 月 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

近年来，社交网络（如推特、微博）被互联网用户广泛使用。但同时此类平台中存在大量不真实谣言，其传播造成的危害巨大。由于社交网络消息数目庞大，使得人工排查谣言成本很高，因此对谣言的自动检测技术具有重大意义。

本文描述了一个谣言检测系统，此系统首先对社交网络中的消息进行谣言特征匹配，检测出疑似谣言的消息；然后对这些候选消息进行聚类，生成疑似的谣言话题；接着对候选话题进行特征抽取，利用监督学习技术对其进行可疑度排名，最终检测出最有可能为谣言的话题。

在对消息进行聚类时，本文采取了基于相似度矩阵的聚类方法，研究了包括发布时间、消息标签、命名实体、文本单词、用户群体的相似度度量，聚类上采用了k均值、层级聚类以及谱聚类等方法，并对比了不同度量和聚类算法之间的优劣。实验表明，此系统能有效地将属于同一话题的不同消息归为一类，降低人工排查谣言的成本。

在对候选话题进行特征抽取时，本文总结相关工作，选取了45种不同的特征，并引入了名为过滤器和包装器的两种特征选择技术，对比两种技术并将其结合使用，最终选择出有效的特征子集。实验证明，经过选择的特征子集能有效提高系统对谣言的识别准确度。

在对话题进行可疑度排序时，系统选取了决策树和朴素贝叶斯两种分类器算法，并引入多分类器组合投票技术。实验表明，引入组合投票技术能有效提高排名系统的实用性，显著降低人工排查的成本。

**关键词**：社交网络；谣言检测；聚类；特征选择；监督学习

ABSTRACT

In recent years, social network (e.g., Twitter and Weibo) has been widely used on the Internet. But at the same time, there are a large number of false rumors in social network. Their spread may have harmful effect on individuals and society. The large amount of messages on social network make the cost of manual checking extremely high. As a result, automatic rumor detection techniques are of great significance.

This thesis describes a rumor detection system. It first filter out suspected rumor tweets by pattern matching, and then cluster these candidate tweets into topics. After that, it extract features of the candidate topics and rank them by the likelihood of false rumor through supervised learning technique. In this way, the most likely rumor topics can be detected automatically.

The system applies clustering techniques based on similarity matrix. The thesis studies similarity metrics that consider different factors including published time, hashtags, named entities, terms and users. The system employs k-means, hierarchical and spectral clustering algorithms. Experiments demonstrate that the system can lower the cost of rumor checking by effectively clustering similar messages into same topics.

For feature extraction, the thesis chooses 45 different features and introduces two feature selection techniques, Filter and Wrapper, to select effective feature subset for the training of next step. A method of combining these two feature selection techniques have been proposed in this thesis. Experiments show that the new method and the step of feature selection can effectively improve the accuracy of rumor detection.

To rank and find most likely rumor topics, the system employs Decision Tree and Naive Bayes as classifiers and a multiple classifiers combining technique by voting has been introduced. Experiments demonstrate that the combining method can effectively improve practicability of the detection system.

**Keywords:** Social network; rumor detection; clustering; feature selection; supervised learning

目录

[第1章 引言 1](#_Toc452860772)

[1.1 社交网络 1](#_Toc452860773)

[1.2 社交网络中的谣言 1](#_Toc452860774)

[1.3 谣言检测 3](#_Toc452860775)

[1.4 谣言检测面临的挑战 4](#_Toc452860776)

[1.5 相关研究概述 7](#_Toc452860777)

[1.6 论文组织结构 9](#_Toc452860778)

[第2章 系统概述 10](#_Toc452860779)

[2.1 原系统流程概述 10](#_Toc452860780)

[2.2 原系统的不足与新系统的改进 12](#_Toc452860781)

[第3章 话题聚类 14](#_Toc452860782)

[3.1 系统采用的聚类算法 14](#_Toc452860783)

[3.2 相似度度量 16](#_Toc452860784)

[3.3 实验数据集与聚类评价指标 19](#_Toc452860785)

[3.4 实验与分析 20](#_Toc452860786)

[第4章 特征选择 31](#_Toc452860787)

[4.1 特征选择简介 31](#_Toc452860788)

[插图索引 32](#_Toc452860789)

[表格索引 33](#_Toc452860790)

[参考文献 34](#_Toc452860791)

[致 谢 36](#_Toc452860792)

[附录 A 外文资料的调研阅读报告 38](#_Toc452860793)

# 引言

## 社交网络

社交网络是近十年来新兴起的一类互联网社交平台，其在国外的代表有推特、脸书（Twitter, Facebook），在中国的代表有微博、人人网等。社交网络，顾名思义是将人们日常的社交活动，推广拓展到了互联网中，或者说是由互联网公司提供一个平台，给互联网用户提供在线交流、分享、交友等服务。

近年来，这类平台的理念、模式日趋成熟，在发展上取得了巨大成功，甚至逐步成为现代人生活的一部分。究其成功的原因，是这类平台将社交这种活动的门槛降低，同时拓展了其形式、主体，大大满足了人类喜爱社交的心理：任何人都能很简单地在此类平台上注册，然后通过简单的操作来发布消息、接收消息、推广消息，用户也能利用平台添加生活中认识的朋友，或者结识素未谋面的新朋友，通过将社交活动移植到互联网，社交网络成功让人们不出门就能完成社交；正是由于社交网络能通过很低的成本拉近人与人的距离，很多文体明星、政治名人（如歌坛巨星、美国总统）都在平台上注册，人们从此可以通过关注他们的账号，与这些原本在生活中很难相遇的名人进行交流、互动，而名人从此也能通过平台很容易地与自己的粉丝分享生活的点点滴滴、对社会事件的看法等等，为自己赢得更高的知名度和更多的人气；同时，社交网络的注册不仅限于个人，任何社会团体（如商业公司、公益组织）也都能注册账号，发布与自己相关的信息，通过平台进行宣传、推广，这就将社交的主体从人与人拓展到了人与团体，甚至团体与团体。

正是由于社交网络的壮大发展，使得这类平台中的信息规模以指数形式爆发增长，每天有越来越多的信息在不同的平台用户间流动、传播，产生各种影响和价值。因此这几年社交网络获得了学术界的广泛关注，越来越多的学者开始研究其中的信息流动模式，或发展相关技术来自动检测有价值的信息和用户，希望能通过研究更好地发挥社交网络的作用，使其产生正面的、更有价值的社会影响。

## 社交网络中的谣言

在现实的社交生活中，存在着一些不真实或真实性有待确定，但却被社交圈中的人们广泛讨论、传播的“小道消息”，学术界称之为“流言”或“谣言”(rumor)。与真正社交活动相似，社交网络中也存在大量的谣言，这些谣言一般分为两种：误传消息和虚假消息。

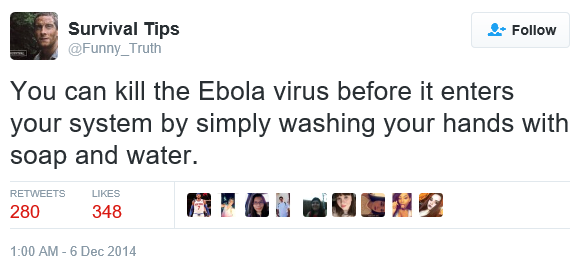
误传消息原本多为源头正规的真实消息，但却在信息的传播过程中被部分用户误解，导致消息在传播中逐渐走样、变形，成为不真实的消息。如图 1.1是推特中的一则误传消息，消息发布于埃博拉病毒盛行的2014年，其大意是我们可以通过用肥皂洗手，简单地杀死手上那些还未进入体内的埃博拉病毒。之所以说这是一则误传消息是因为消息本身并无恶意，是想鼓励更多的人勤洗手来预防埃博拉病毒，事实上这也是世界卫生组织一直提倡的；但是此消息的内容却不具有真实性，原因是用肥皂洗手可以减少手上附着的病毒数目，却不一定能简单地杀死埃博拉病毒（参考信息来源：http://www.killebolavirus.com/does-hand-sanitizer-kill-the-ebola-virus/），而世界卫生组织也从未提到一般的肥皂能有效杀死埃博拉病毒。

图 1.1 推特中的误传消息

（图片来源：https://twitter.com/funny\_truth/status/541155156264382465）

虚假消息，则是另一种更加普遍、潜在危害更大的谣言。这种谣言的特点是其发布时并无可靠的消息来源或有力的证据进行佐证，通常是人为臆测或刻意捏造产生的，其内容不具有真实性且常带有恶意或中伤。但通过包装，此类消息能获得很高的隐蔽性，虚假性很难被人们一眼辨识出来；通过包装，此类消息也能具有很强的轰动性，使得很多用户不顾其是否真实，盲目地争相转发。由于社交网络的用户量庞大、用户网络连通性强，虚假消息很容易被广泛传播：从一开始仅有的零星几个源头，发展成被成千上万人知晓、推送只需要短短几个小时。一旦这类消息被广泛传播，将有可能对某些个人或者团体的利益造成巨大损害，甚至引起社会的恐慌，产生非常负面的社会影响。如图 1.2这则推文谈论的就是一则虚假消息：当时谣言盛传美国摇滚歌手Akon在刚果共和国的一场庆典中将自己包裹在一个巨型的泡泡球内进行表演，其目的是为了避免自己在与群众接触时感染到当时在非洲盛行的埃博拉病毒。这则谣言当时在推特上疯传，部分用户对谣言表示怀疑、不确定（如图 1.2），也有Akon的粉丝站出来为他说话，但也有很多用户因此对Akon的为人表示愤怒、谴责和质疑，这使得其名声受到了很大的冲击。但后续的相关报道证实这只是一则人为臆测的虚假消息：Akon这种表演形式只是为了和观众互动，远在几年前埃博拉病毒未爆发时，他就已经用过同样的形式在世界各地进行表演了（参考信息来源：http://www.independent.co.uk/arts-entertainment/music/news/akon-didnt-perform-in-a-bubble-in-dr-congo-because-of-ebola-he-was-just-having-a-fun-time-for-9772004.html），原谣言缺乏有力论据。

图 1.2 推特中的虚假消息

（图片来源：https://twitter.com/realalexjones/status/517808892437204992）

由于社交网络每天都产生并传播着大量的消息，其中混杂了不少诸如此类的谣言，如果不加以监督管制，尽早发现危害性高的谣言，及时对其进行辟谣并将谣言散播者绳之于法，那么社交网络将成为虚假消息传播的温床，将对个人、团体以及社会造成不可估量的负面影响。

## 谣言检测

因为社交网络存在不少拥有负面影响的谣言，所以必须找到一种有效的监管机制。这就面临着一个很重要的问题：如何找到社交网络中的谣言？

一种朴素的想法是设立“消息审查员”的职位，利用人力去审核推特上的消息，找到其中疑似的谣言消息。但是社交网络的用户规模、消息数量巨大，若要对消息进行逐条的人工审核，那么需要的审查员数量也将极大，其人力成本将极高。因此人工地进行逐条审查并不现实，这不是一个合理有效的解决方案。

正是因为人工检测成本太大，所以研究学者们转而研究开发能自动识别、检测出谣言的技术。这类技术通常会利用计算机强大的处理能力分析海量消息，分析其文本、发布者或传播路径等等的特点，自动找到可疑度高、疑似谣言的消息。但由于机器识别的准确率有一定的限制，所以机器识别出来的可疑消息通常还需要送与审查员进行人工复查，确认为谣言才能采取后续的辟谣和法律相关工作。

以上是当前被广泛认可的，监管社交网络中谣言传播的解决方案之一，学术界称之为谣言检测（rumor detection）。目前还存在一些其他的解决方案，如设立辟谣公众账号，一旦群众用户发现疑似谣言的消息，就举报给公众账号，由公众账号选择举报度高的消息进行审查、辟谣；但本文只专注于谣言检测而略过其它。

本文所指的“谣言检测”，是专指从社交网络的消息中，自动检索出谣言（包括误传消息和虚假消息）相关消息的技术。其中“相关消息”是指对谣言进行传播、讨论、怀疑、质问、辟谣的消息。而有关谣言源头分析、谣言传播轨迹追踪等等的其它相关研究，都不属于本文讨论范围之内。

## 谣言检测面临的挑战

谣言检测技术的研发面临了许多问题和挑战，以下是其中两个主要的挑战：识别准确度不高，检测重复率过高。

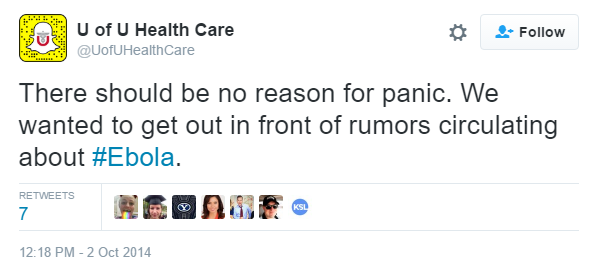
识别准确率不高，是指很多非谣言的消息被机器错误地识别为谣言，或者许多真正的谣言被机器漏过，误认为是普通消息。造成这个问题的因素很多，最主要的一个是分析自然语言难度很高，目前的技术通常只能对文本中的词或者短词组进行语义、情感分析，却很难将文本形成一个整体去分析句子或段落的真正意义。如图 1.3就是一个例子，在这则推文中，存在着如恐慌（panic）、谣言（rumors）和流传（circulating）这些词语，如果仅仅分析它们的语义和情感导向，机器很容易将这则推文误认为是谣言相关的消息。但事实上在恐慌这个词前面还存在没有理由（no reason）这个词组，在谣言和流传这些词前面还存在走出（get out）这个词组。只有将这些词组都串联起来考虑，这个句子的真正意义才能被分析出来：它不是在讲一引起恐慌的谣言，而是在讲我们不应当恐慌，而应该走出来面对那些传播的谣言，这是一则普通消息。

图 1.3 非谣言例子：自然语言分析面临的困难

（图片来源：https://twitter.com/UofUHealthCare/status/517755407100420096）

除此之外，即便自然语言的语义能被机器分析得十分透彻，也不一定能准确地判定一则消息是不是谣言，因为谣言的判定要求对消息的来源、内容的论据进行深入的考证，这就涉及到一定的专业知识和技巧，这不是单一的文本分析就能完成的事情。如图 1.4中的推文称感染了埃博拉病毒的护士Nina Pham的男朋友也被检测出埃博拉病毒的症状，在推文后面作者还附上了佐证的新闻网址，点开后也的确是一则符合推文内容的网页新闻。现在你能告诉我这是一则新闻推广还是一则谣言吗？或许有的读者会认为是前者，因为其附上了消息来源，是一个正规的新闻网站。但事实上我们并不能这么简单地下结论，因为即使是正规的新闻网站也会由于监管不力或为了博取点击率等原因出现一些不实的报道。很遗憾，这则新闻正是这样的报道，后续已有大量媒体对这则谣言进行了辟谣（参考信息来源：http://www.ibtimes.com/ebola-nurse-nina-phams-boyfriend-rumored-admitted-hospital-ebola-symptoms-alcon-releases-1707586）。这个例子说明，光是文本分析并不能完成谣言检测的任务。

图 1.4 谣言例子：谣言判定面临的困难

（图片来源：https://twitter.com/Jody\_Arrington/status/523633560779907074）

因此为了提高谣言检测的准确度，许多研究学者不仅考虑文本语义方面的因素，还综合考虑了如消息发布者信誉度、消息来源可靠度、消息传播路径拓扑结构等等的一些特征，在多元特征上加之以复杂的分析模型（如规则模型、概率模型），最终形成较为鲁棒、识别准确率较高的谣言检测系统。但这其中面临的问题就更多了，例如：如何找出那些有价值的、需要被考虑的特征？不同的分析模型各有优劣，它们之间能否形成互补？本文在接下来的第4章中也在尝试解答以上的两个问题。

检测重复率过高，是谣言识别技术面临的另一个挑战。现在，假设我们的谣言检测系统的识别准确率达到了100%（即便这是不可能的），即便如此，但因为社交网络中的消息数目极大，经常会有成千上万个用户在讨论同一个话题，假设他们都在讨论、推送同一则谣言，这些消息自然都能被我们卓越的系统成功地检测出来。但当我们将检测出来的谣言交给审查员进行人工复查，或者交给相关部门进行辟谣和法律维权时，问题就出现了：这么多的谣言消息，审查人员和相关部门根本看不过来，最重要的是，有大量的消息都是关于同一个谣言的，重复去审查、处理这些消息根本毫无意义。要想利用有限的人力资源审查更多不同的谣言，就必须将那些讨论同一话题的消息归为一类，降低检测的重复率。

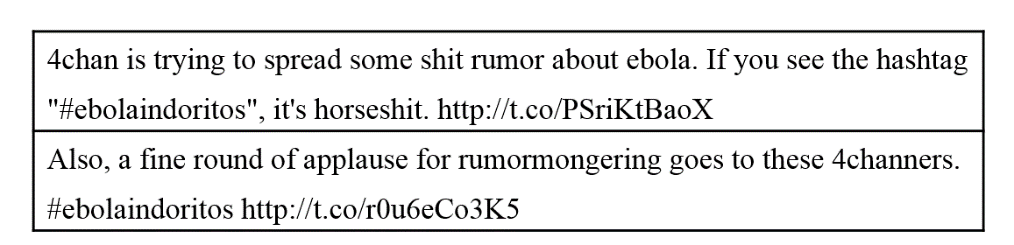
降低重复率，实际上是一个文本聚类的问题。对文本聚类的相关研究已不少，但很多传统的聚类方法对社交网络的文本不适用或不足够准确，因此还需要开发出符合社交网络文本特点的聚类方法。这个问题的主要挑战在于社交网络的消息通常都是短文本，其文字信息不足以完成聚类。如图 1.5的两则推特消息都在讨论同一则谣言：当时在美国的图片网站4chan上盛传一家零食公司Doritos的零食中存在埃博拉病毒。虽然这两则消息在讨论同一谣言，但其文本相似度却很低：由于文本短小，交叠的关键词本就不多，有4chan、rumor、ebolaindoritos，但是在第二则消息中4chan却被表达成4channers，而rumor则与mongering连在一起导致成为了另一个词，所以重复的关键词仅剩下了ebolaindoritos一个；另外文本附上的超链接地址完全不同，尽管它们指向的新闻报道内容相似。如果从传统的聚类方法出发，这两个文本很可能被认为不属于同一类。那么如何才能将社交网络的文本准确聚类呢？研究者们通常会在传统方法的基础上，结合考虑一些社交网络特有的要素，如消息发布时间、消息话题标签、消息提及用户等等。而又因为很多特有要素（标签、提及）不是每个文本都有，所以对应的相似度矩阵经常是稀疏的，所以对聚类方法的选择上也有一定的讲究。本文第3章中将对这个问题进行详细阐述。

图 1.5 两则讨论同一个谣言的消息

尽管谣言检测面临着以上各种挑战，但相关的研究也一直在致力于克服它们。

## 相关研究概述

由于社交网络的兴起是近十年的事情，所以针对它的谣言检测技术目前仍在发展阶段，相关研究不多，大致可以分为两类：监督学习技术、筛选排名系统。

以监督学习技术为主导的谣言检测技术，一般致力于提出针对社交网络消息，适用于谣言识别的有效特征，然后使用带有标记（是否谣言）的社交网络消息数据集，引入各种分类器进行学习、训练，或提出新的分类器模型，用交叉验证的方法评估、找出准确率更高的谣言检测方案。

早期的这类研究选取的消息特征较为简单，大致分为内容特征、用户特征、传播特征[5,10]：内容特征考虑了词语、多元词组，以及社交网络特有的超链接地址（url）、话题标签（hashtag）和提及符号（@[username]）；用户特征考虑了消息发布者粉丝数量、关注数量、影响力、发布消息总量、注册时间、发布消息地区等等；传播特征则考虑了消息是否原创、评论数量、转发数量等。

而后续的研究陆续提出更有效的特征。Shengyun Sun等人的研究[7]加入考虑了多媒体要素，提取社交网络消息中附带的图片，借助搜索引擎评估图片的可疑程度，形成新特征；Carlos Castillo等人的工作[2]虽然时间也很早，但考虑特征高达68类，比前文的基础特征多考虑了标点符号、文本情感评估、消息传播树、话题分布特征等等；Sejeong Kwon等人的研究[4]提出了一种基于时间序列分析的特征，用名为PES的模型拟合消息转发数目与时间的关系，并学习出峰值强度、峰数目、峰周期等10个参数作为新特征；Ke Wu等人的研究[9]则是提出了一种图核（graph kernel）的度量作为新特征，他们将用户分为普通用户和有影响力用户两类，对消息抽取其在这两类用户间转发的传播树，并提出一种基于随机游走的算法计算树的相似度，最终形成新特征。

而在分类器方面以上研究选择各不相同，一部分选用了较为简单的分类器：采用决策规则的研究[2]、采用朴素贝叶斯的研究[5,7]、采用线性回归的研究[5]、采用支持向量机的研究[2,4,9,10]、采用决策树的研究[2,4,7]；也有的研究使用了较为复杂的分类器：采用贝叶斯网络的研究[2,7]、采用随机森林的研究[4]、采用神经网络的研究[7]。以上研究根据消息数据集的不同、特征选取方案的不同，表现最好的分类器也各不一样，目前仍没有定论说哪种分类器最好，也缺乏理论指导在什么类型的数据集中采用哪种分类器更好。

以筛选排名系统为核心思想的谣言检测技术，不是对每条消息直接进行分类，它们首先对消息数据集进行话题提取，然后对同一话题中的消息再抽取统计特征，最后设计出一套可疑度评估规则，或是引入可求后验概率的分类器，对消息话题进行可疑度排名，筛选出最可能是谣言的话题列表。

Tetsuro Takahashi等人的研究[8]提出了一套简单的谣言筛选机制，总共有三个步骤：一，对消息数据集的文本进行命名实体(named entity)抽取，对抽取的关键词做爆发检测（burst detection），找出讨论频率高的话题；二，对话题相关的消息集计算转发率，设置阈值筛选出消息转发率高的话题；三，对话题消息集统计其含有谣言线索词（如“false rumor”）的消息比例，设置阈值筛选出那些含谣言线索词比例最高的话题，作为最疑似谣言的话题列表进行输出。这套系统使用的技术和筛选机制过于简单粗糙，其检测的准确率和召回率也势必不高。但他们的思路值得借鉴：不再是对每条消息进行谣言判定，而是先检测出重要的话题，再检索相关消息，对话题内的所有消息进行统计特征的提取；不再是判定是否谣言，而是对消息话题进行可疑度排名；不再纯粹用分类的思路，而是引入了线索词的概念，尝试用文本匹配筛选出谣言消息。

Zhe Zhao等人的工作[11]借鉴以上思路并进行拓展，设计出了一套更加完整、鲁棒的筛选排名系统。该系统分为五步：一，通过模式匹配检测出谣言信号消息；二，谣言信号消息聚类；三，对信号消息进行关键词组提取；四，利用关键词组比对将普通消息归到信号消息类中；五，使用监督学习技术对每个消息类进行可疑度排名。该工作论证了他们系统在时间上具有高效性，而且能很早地检测出那些刚开始传播的谣言，这两个特点使得该系统能处理大规模的消息数据，还能用于及时阻止谣言的进一步传播，非常满足社交网络的真实需求，实用性高。事实上本文的系统正是在此系统的基础上进行提高、拓展，因此在第2章中还会再详细介绍此系统的流程架构以及本文所做的改进。

更详尽的相关工作介绍可参考附录A。

## 论文组织结构

本文剩余部分将按照以下结构进行组织：第2章将介绍本文实现的谣言检测系统概况和关键改进；第3章将介绍系统改进中实现的二次聚类技术与其效果；第4章详细介绍各类特征选择技术，并新提出的一种特征选择方案；第5章将简要介绍本系统采用的监督学习技术和组合投票技术，并通过实验比较不同的特征选择技术和分类器选取对系统检测准确率的效果，论证了新特征选择方法和组合投票技术对系统的提高作用；第6章将进行总结并介绍未来的工作。

# 系统概述

## 原系统流程概述

本文实现的谣言检测系统是基于一项已有工作的谣言筛选排名系统[11]，对其进行拓展和改进，本小节将对原系统的检测流程和设计细节进行介绍。

图 2.1 原系统流程图

（图片来源：参考文献中Zhe Zhao等人的工作[11]）

原系统的流程图见图 2.1，一共有五步，下面逐步进行介绍。

第一步，信号消息识别。该工作所指的“信号消息”（signal tweets）是指那些包含怀疑、惊讶、质疑、辟谣文字的消息，他们认为任何谣言在传播过程中，都一定会受到一部分明智用户的怀疑或质疑，而随着时间的推移也会有用户出来辟谣，因此谣言检测可以从检测这些“信号消息”出发，提取这些消息讨论的话题内容，就是潜在的谣言。而原系统识别信号消息使用的是文本的模式匹配，图 2.2就是他们使用的匹配模式的正则表达式和类型。由于社交网络消息为短文本，文本模式匹配耗时低，即便是对规模很大的消息数据集检测信号消息，其速度也非常快。当然，符合模式的消息中有很多谈论的并不是谣言（例如可能是令人难以置信的真实新闻），所以还需要后面的步骤进一步识别。

图 2.2 原系统使用的匹配模式

第二步，信号消息聚类。对上一步检测出来的信号消息进行聚类，原系统使用的是基于距离阈值的聚类方法，距离函数使用Jaccard相似度（一种广泛用于计算集合相似度的度量），具体方法如下：对需要计算距离的两消息，分别提取文本的单词、双单词词组以及三单词词组形成各自的一个集合，最后对这两个集合计算Jaccard相似度；在聚类过程中对于两个消息，如果其集合的Jaccard相似度超过预定义的阈值，则归为一类。也就是说：1、如果消息a被归入类S，那么S中至少存在另一则消息b，满足a与b提取出来的集合相似度超过该阈值；2、如果S1与S2是不同的类，那么对于S1中的任一消息a和S2中的任一消息b，它们提取出来的集合相似度都不超过该阈值。这里每个信号消息类表示一个话题。

第三步，对信号消息类提取关键词组集。该工作中的关键词组集（statements）的集合元素是那些在信号消息类中出现频率超过预定义阈值的单词、双单词词组和三单词词组。关键词组集的元素将会作为该信号消息类的话题关键词。

第四步，归类非信号消息。这一步的目标是将整个消息数据集中的那些非信号消息尝试归到某一个信号消息类中形成完整的消息类（也允许不归入任何类）。使用的方法与第二步类似，但利用了关键词组集：对于一则非信号消息，先提取它的单词、双单词词组以及三单词词组形成的集合，然后遍历所有信号消息类，计算该集合和信号消息类的关键词组集的Jaccard相似度，如果超过预定义的阈值则将该非信号消息归入此消息类，否则就去看下一个信号消息类是否满足条件；如果所有信号消息类都不满足条件，则该非信号消息归类失败，不列入消息类中。这一步产生的所有消息类（每类必含信号消息，可能含非信号消息），将作为筛选后产生的谣言候选话题，进入最后一步的可疑度排名。

第五步，消息类可疑度排名。对第四步中产生的每一个候选消息类，提取其统计特征（包括信号消息比例、转发消息比例、消息平均长度、消息平均含超链接数目、含话题标签数目、含提及符号数目等），将这些特征输入分类器，利用有标数据集（谣言消息类vs非谣言消息类）进行训练。原系统选择了决策树作为分类器，因为它不仅能对某一消息类进行分类，还能输出一个类似“后验概率”的值，表示该消息类属于谣言话题的可能性。将训练后的分类器对所有消息类计算它们是谣言的“后验概率”，按此值对消息类进行排序，找出可疑度最高的N个消息类，作为系统最终检测出来的谣言话题进行输出。

以上就是原系统完整的流程步骤，本文首先按此进行了完整的实现，然后针对原系统的局限进行了拓展和改进，形成新系统。下一小节将详细阐述。

## 原系统的不足与新系统的改进

在原系统的文章中论证了该系统的高效率，以及其用于谣言防治的实用性。但该系统仍存在以下两个比较显著的问题：

一，检测出的话题重复率过高。由于原系统为了提高速度，在聚类上使用了较为简单的阈值法并设置了较高的阈值，而在距离函数上则选取了计算重复率的Jaccard相似度，因此原系统中每个类内部的所有消息（包括信号与非信号的），其文本内容的重复率都特别高，经常是几乎一模一样的句子，只差了几个单词。这导致的问题是，系统输出的前N个疑似谣言的消息类中，有很多其实谈论的是同一个话题，但是原系统却不能把它们聚成一类，原因只是两类中的文本词组重叠率低。这使得如图 1.5这样的两则消息被归到了两类当中，尽管它们在讨论同一个话题。此问题在原系统的结果集中非常普遍，虽然看上去无伤大雅，但在实际应用中，由于还需要将筛选出来的疑似谣言话题交予人工复核，设想审查员反复看到的几十个候选话题其实都属于同一个话题，那么复核效率必然大大降低。

二，检测准确度不高。通过模式匹配来发现信号消息的思路很好，但是这样检测出来的话题必然有很多不是谣言，因此后续的可疑度排名就尤为重要。但原系统只考虑了非常简单朴素的13个统计特征，特征的多样性不足会导致无论采用怎样的分类器都不能获得很好的分类排名效果。因此可以说原系统在可疑度排名这一步还有很大的改进空间，这将是提高谣言准确度的关键之一。

根据原系统以上的不足，本文对其进行了针对性改进，方案如下：

一，在原系统聚成的消息类的基础上，再做一轮二次聚类，目的是将讨论同一话题的不同消息类重新聚成一类，降低检测结果的话题重复率。

二，为消息类抽取更多种类的特征，并引入特征选择技术，目的是通过增加特征多样性和选择有效的特征，增加系统的谣言检测准确率。

以下对上述方案稍作解释：

1、为什么不直接更换原系统的聚类算法，而是做二次聚类？这是因为原系统的聚类算法虽然粗糙但非常高效，通过提取信号消息类的关键词组集（statements）以及使用Jaccard相似度进行类与消息比对的方式，不尽降低了复杂度且没有反复迭代的过程，能高速地将那些内容高度相似的消息聚在一起，这样也能做到基本肯定每个消息类内的所有消息都是讨论同一个话题。因此我们没有必要更换原系统的聚类算法，而是应该直接在其聚类结果上再进行聚类。

2、为什么在特征上改进，而不是在分类器上改进？为什么要引入特征选择技术？不在分类器上改进是因为目前没有定论哪种分类器效果更好，也不是越复杂的分类器的检测准确率就越好，这跟特征选择和消息数据集都有关，因此分类器的选择只能通过尝试。然而原系统的输入特征少是硬伤，无论是怎样的分类器，如果输入的特征种类太少且不够有效，那么都无法达到很好的分类效果，所以在特征上改进是需要优先完成的。至于为什么引入特征选择技术，是因为目前相关研究提出的特征种类繁多但同样没有定论说哪些特征比其它有效；而如果一股脑地将这些特征全部输入分类器，那么其中那些低效、冗余的特征会影响检测效果，而且很有可能会因为训练数据集样本不足而造成训练过拟合，反而降低分类器准确率。因此需要选择高效紧凑的特征子集，通过保证特征质量来提高分类器的分类排名效果，最终提高系统的检测准确率。而特征选择技术能解决这个问题。

下面两个章节将分别介绍这两方面的改进：第3章介绍系统实现的二次聚类算法与其带来的成效，第4章介绍系统引入的特征选择技术和新特征。而在第5章将简要介绍新系统采用的分类器和相关技术，通过实验比较不同特征选择技术和分类器组合对新系统检测准确率的影响。

# 话题聚类

## 系统采用的聚类算法

聚类（clustering），是要完成以下这样的任务：将一组对象中的每个元素归入多个组内，使得相似的元素在同一组，而不相似的元素在不同的组，聚类后的每个组被称为一个类或簇团（cluster）。

关于聚类的研究目前已有很多，研究者们也提出了很多不同的聚类算法。由于本系统处理的对象是社交网络中的消息，聚类需要对消息提取特征，然后将特征相似的对象归为一类，但不同的特征（如时间特征和文本长度特征）数值化之后难以统一度量单位，所以即使能将特征表示为特征空间中的向量，此特征空间也不是欧几里得空间。所以一些依赖于欧几里得距离度量的聚类算法，如基于密度的聚类算法，或基于网格的聚类算法都不太适用。在这种情况下通常的解决方案是根据对象特征的特点，设计一个合理的相似度度量公式，通过这个公式计算数据集中对象两两间的相似度或相离距离，生成一个相似度矩阵，然后根据相似度矩阵进行聚类。

可以利用相似度矩阵进行聚类的算法也有很多，本系统挑选采用了以下几种:

K均值聚类法[12]，是一种通过不断更新聚类中心完成聚类的迭代算法。由于该算法需要计算对象特征点与聚类中心的距离，还要计算同一类中所有特征点的聚类中心，所以通常此算法也依赖于欧几里得特征空间。但是此算法也能利用相似度矩阵进行聚类：相似度矩阵中的每行代表一个对象与其它所有对象的相似度（通常值域为[0,1]），这里可以将每一个相似度看成对象的一个特征，将整行看成是该对象的特征向量，或其在特征空间中的特征点。由于将此特征空间看成欧几里得空间具有合理性，所以可以使用K均值聚类法进行聚类。之所以说看成欧几里得空间是合理的，是因为特征空间中的每一维都代表一个对象（准确说是其它对象对此对象的相似度），由于数据集中的对象是平等的，所以特征空间中的每一维都是平等的，而且每一维的度量都是统一的（相似度在[0,1]之间），其距离的意义也对于聚类也是合理的（如果两特征点在某一维比较接近，说明对应的两个对象都与某个对象比较相似或都不相似，那么它们本身也可能相似；如果两对象在某一维不接近，说明其中一个对象与某对象相似，而另一个对象与这个对象不相似，那么这两个对象本身也很有可能不相似）。K均值算法的优势在于能将特征点聚类成“球状”，使得类内部对象的特征点两两之间都很接近。

基于核函数的K均值聚类法[13]，算法流程与K均值聚类法相似，但在聚类之前需要将特征点通过一个非线性函数（核函数）投影到高维空间中，目的是将原本线性不可分的特征点变成线性可分或近似线性可分，然后在此高维空间中进行K均值聚类。常用的核函数有多项式核函数、高斯核函数以及S型核函数等，本文系统尝试使用高斯核函数进行聚类。同理于K均值聚类法，基于核函数的K均值算法也可以利用相似度矩阵进行聚类。

层级聚类法[14]，是一种基于距离矩阵的聚类算法，通常分为自顶向下和自底向上两种，本文采用了自底向上的层级聚类法。算法的核心思想是贪心地将两个距离最近的对象聚为一类，成为一个新的“对象”，然后再加入对象集中继续贪心地迭代聚类。由于经过迭代后新的“对象集”中的元素不再是单个对象，而是一组对象，所以在计算新的元素间的距离时可以有不同的计算公式。本系统尝试使用的距离计算公式有：简单连结（single linkage），组间距离定义为组间对象的最短距离；完全连结（complete linkage），组间距离定义为组间对象的最长距离；平均连结（average linkage），组间距离定义为组间对象的平均距离；加权连结(weighted linkage)，计算组内对象的加权质心，将组间距离定义为质心距离；沃德方法（Ward’s method），并不是贪心地找到距离最近的两个组合并，而是找到这样的两个组：它们合并后能使得总共的组内均方差和增量最小（minimum variance criterion），将这样的两个组合并。不同的距离计算公式会影响聚类效果，这也与数据集的数据特点有关，所以在使用层级聚类算法时应尝试不同的公式；而算法中需要用到的距离矩阵，可以通过常量矩阵减去相似度矩阵得到。层级聚类法的好处在于其聚类速度快，且保证了任一对象至少与同类的某些对象高度相似。

谱聚类法[15]，是一种基于相似度矩阵的聚类算法。算法通过计算相似度矩阵的特征向量和特征值，抽取特征值最大的前n个特征向量拼接成新的特征矩阵（相当于降维，将每个对象的特征数量变成n个，形成n维特征空间），然后再利用特征矩阵进行聚类（如使用K均值聚类法）。此算法的好处在于通过抽取主要的特征向量能对相似度矩阵进行降噪，降维后的特征空间理论上会使聚类效果更好。

以上四种聚类算法均能利用相似度矩阵将对象聚类，都可以尝试用于系统当中，而实际聚类效果如何，还需要实验来评估比较（参考本章最后一小节）。但是在在实验之前，我们可以预期哪种聚类方法比较适合于社交网络中的话题聚类。由于社交网络的消息是不断流动和传递的，通常用户是在看到别的用户讨论某个话题后，对此话题进行评论或推广，所以理论上每个用户的消息都至少与另一个讨论同话题的用户的消息有一定的相似性，这个相似性可以是表现文本内容上（如关键词相似），也表现在时间维度上（如发布时间接近）。因此，上述介绍的聚类方法中，层级聚类应当比较适合解决社交网络的话题聚类问题，因为它可以保证任一对象至少与同类的某些对象高度相似。当然，由于社交网络讨论同一话题的消息集，其源头通常不止一个，而且也有很多用户的消息是原创的，所以层级聚类法并不能保证可以将讨论同一话题的所有消息聚成一类。

## 相似度度量

仅是有上一小节的聚类算法，聚类还不能实施，我们仍缺少关键的相似度矩阵。想拥有良好的聚类效果，就需要针对数据集的特点设计出合理的相似度度量公式，这一小节将探讨什么样的相似度度量适合于社交网络的话题聚类。

时间相关的因素。本文在上一小节中提到，由于社交网络中大部分消息是通过推送传播的，所以很多情况下是用户看到了另一用户的消息后进行评论、转发，而社交网络中关注者通常都能在第一时间内看到被关注者发布的消息，而且立刻作出回应，因此讨论同一话题的消息很可能在发布时间上是非常接近的。基于这个想法，本文提出社交网络的话题聚类应当引入时间相似度度量。因为社交网络中不经常出现被关注者发布消息后，过了很久才被关注者看到并回应的情况，所以我们可以认为：只有两消息的发布时间间隔很短，两消息才有较高的时间相似度；一旦时间间隔稍长，可以认为两消息基本没有时间相似度。因此，发布时间间隔到相似度的函数可以使用指数衰减函数，如下：

此公式中是两则消息的发布时间间隔，而是一个预定义的参数，类似于物理学中的“半衰期”，也即时间间隔如果等于此值，则时间相似度衰减为0.5。由于社交网络消息的时效性，通常将设为不大的值，在本系统的实现中将其设为1天，而实际上若想更加突出时效性甚至可设成半天或几小时。如果是对两消息类求时间相似度，则先求出分别的平均发布时间，然后用它们来计算时间间隔。

用户互动相关因素。其出发点与时间相关因素相同，由于用户经常是看到另一用户的消息后转发，而转发消息是带提及标记的（在推特中是“RT @username”，其中@是提及符号），那么如果某消息中含有提及符号，其提及的用户发布了另一则消息，那么有可能这两则消息在讨论同样的话题；同样，当两则消息在谈论同一话题，它们之间或许不是转发与被转发的关系，但很有可能会同时提及（@）话题相关的另一个用户（比如谣言中的主角），从这个角度出发，两则提及了同一个用户的消息，以及被提及的用户所发布的消息（比如谣言主角进行解释、辟谣的声明），有可能都是在讨论同一话题。以上两种场景都是社交网络中特有的用户互动，这种行为能给话题聚类提供相似性的线索，因此可以考虑将其加入相似度矩阵中。在本系统的实现中用户互动相似度的度量公式如下：

可以看出，在本系统中用户互动相似度是二值的，这样设计的原因是提及关系在社交网络中非常稀疏，一旦出现则可以认为两消息关联性很高。在系统实现中，如果求的是两个消息类的用户互动相似度，系统会对两消息类分别求出提及用户名列表和作者列表两者组成的用户名集合，如果两消息类的用户名集合交集不为空，则相似度为1，否则为0。

文本内容相关因素。出发点很自然：谈论同一话题的消息，其文本内容可能也非常相似。本系统一共考虑了四种文本内容相关的相似度度量。

话题标签相似度。话题标签（hashtag）是社交网络中一种独有的元素，社交网络允许用户在消息文本中附带话题标签（推特中是“#topicname”），通过这样来将消息进行话题归类，便于其他用户进行检索和阅读。因此非常自然地，如果两则消息被打上了相同的话题标签，那么它们很有可能谈论相同的话题。在系统实现中是统计两消息（或消息类）分别的话题标签集合，然后计算两集合间的Jaccard相似度作为话题标签相似度。

命名实体相似度。命名实体（named entity）的概念有点类似专有名词，一般是指文本中的人名、组织名和地名等[16]。由于微博上的讨论话题（如谣言）经常为事件，一般会含有事件主体、地点等要素，如果此类型的命名实体在两则消息中同时出现，那么这两则消息很可能讨论的是同一话题。因此加入命名实体相似度，理论上对社交网络的话题聚类有一定的帮助。具体实现上可以利用命名实体识别技术（named entity recognition），提取两个消息（或消息类）分别的命名实体集合，然后计算两集合间的Jaccard相似度作为命名实体相似度。

文本单词集的Jaccard相似度。此相似度出发点非常朴素：谈论同一话题的消息，其文本可能是相似的，那么其出现的单词也应该有很高的重叠率。事实上社交网络中谈论同一话题的消息的确有很大部分属于转发消息，而转发消息的文本重叠率非常高；即便不是转发，因为两消息的作者可能是看了同一消息源后进行意见发表，那么其措辞和关键词很可能会受到消息源文本的影响而导致相似。因此，文本单词集合的Jaccard相似度在这些场景下对聚类有很大的帮助。其符号记为。

文本词频-逆文档频率相似度。词频-逆文档频率（tf-idf）是一种统计的权重，反应一个单词在文本集的某文本中的重要程度[18]。其由两部分相乘得到，第一部分是词频（term frequency），表示单词在文本中的出现频率，反应的是单词在文本的重要程度；而第二部分是逆文档频率（inverse document frequency），表示含该单词的所有文本，占文本集中所有文本比例的倒数，反应的是单词在文本集中的独特程度。此概念被广泛使用于文本检索中的相关性排名，引入此概念的出发点是前三个相似度类似，实际上都是认为讨论同一话题的消息的文本内容有相似性。直接取单词集合的Jaccard相似度太过简单，一些常用单词（如is、the、a）在任何文本都可能出现从而其重叠率不能代表内容相似度，我们真正需要的是关键词的重叠率，但如何定义“关键词”呢？关键词可以有很多，它可以是话题标签，也可以是命名实体，而词频-逆文档频率则是引入逆文档频率的概念，将定义关键词的任务自动化，把那些在文档集中出现频率不高的独特单词看成关键词。虽然这样的定义不完全正确，但是有一定的道理，而且多年来的实验证明词频-逆文档频率的确是非常有效的单词权重计算方式，所以在社交网络的话题聚类中引入文本词频-逆文档频率相似度是非常推荐的。在具体系统实现上，先对两消息（或两消息类）统计出现单词的词频和其逆文档频率，计算分别的词频-逆文档频率特征向量，然后计算两向量的余弦相似度作为词频-逆文档频率相似度。

以上介绍的6种相似度度量，包括1种时间相关相似度，1种用户互动相关相似度，以及4种文本内容相关相似度，它们用于话题聚类都有一定的合理性，但是仅仅是其中1种相似度高远不能说明两消息讨论的是同一话题，真正的情况是有越多种类的相似度高，两消息属于同一话题的可能性就越高。因此在实际应用中我们应同时考虑这几种相似度，得到一个综合的相似度度量。本系统采用的方式是将这几个相似度加权平均得到综合的相似度：

(1)

由于并不知道怎样的权重分配能得到比较好的结果，所以实验中使用了网格搜索（grid search）的方式遍历参数，确定最好的权重配比。

当然，除了加权平均，也可以通过相乘，或者加乘结合的方式将不同的相似度度量结合考虑。但经过尝试，在本系统采用的数据集中是加权平均的方式最为有效，因此系统最终实现的、以及实验中汇报的就是这种结合方式。下面进行实验汇报，评估不同的聚类算法与相似度度量对社交网络话题聚类效果的影响。

## 实验数据集与聚类评价指标

在进行实验汇报之前，先简要介绍本文使用的实验数据集和聚类评价指标。

本文使用的实验数据集是推特消息数据集，该数据集使用了推特API，以“ebola”（埃博拉病毒）作为关键词进行检索。完整的数据集包括16,711,671条推特消息，共1,240,415个用户，时间范围是从2006年12月25日到2016年2月21日。完整数据集将用于第4章节特征选择和第5章节监督学习技术的实验中，而此章话题聚类的实验将用其在2014年11月的消息子集。此推特消息子集共有1,594,572条推特消息，共274,339个用户，时间范围是从2014年11月1日到2014年11月30日。之所以用此消息子集是因为评价聚类需要对消息人工地打上真实聚类标签，而使用完整的数据集则标注工作量太大，于是使用子集。而推特上热烈讨论埃博拉病毒的时期是2014年9月到2014年11月这三个月，而11月是埃博拉讨论的收尾期，数据量适中，因此选了这段时间的消息子集。

实际上，并不是给消息打上真实的聚类标签。因为本章节的目的是给原谣言检测系统检测出来的谣言候选消息类再次进行二次聚类，降低谣言的话题重复率，所以实际上是给谣言候选消息类打上真实的聚类标签（将讨论同一话题的消息类打上相同的标签）。而原系统在经过模式匹配和聚类后，从2014年11月的埃博拉消息子集中筛选出9,488条谣言候选消息（包括信号消息与非信号消息），原系统聚类后共产生939个谣言候选消息类，只需要对它们进行标注即可。经过人工标注，939个谣言消息类中含有真实话题686个，此子数据集称为数据集A。数据集A之所以有这么多话题是因为数据集中夹杂着很多传播仅限几个人的小范围消息，这些消息经常是一些独立的话题，也就是它会自成一个话题类，因此有很多聚类中实际仅有一个元素，这些消息即使是谣言也并没有得到广泛传播。

为了去除这些小范围消息的噪声话题，本文过滤掉了那些包含消息少于10条的“冷门话题”以及它们的消息，过滤后谣言候选消息剩下8,373条，原系统聚成的谣言候选消息类共有214个，人工标注后其真实话题有67类，也即话题重复率高达3.2倍，换言之原系统的确存在话题重复率过高的问题，这将导致后期人工复核成本的增加（详见章节1.4和章节2.2），因此的确有二次聚类的需求。此子数据集称为数据集B。

聚类评价指标。本文采用的聚类评价指标是被广泛使用的归一化互信息[19]（normalized mutual information (NMI)），这是一个值域为[0,1]的指标，越高的NMI值表示聚类效果越好。

## 实验与分析

本文分别在数据集A（埃博拉消息集在2014年11月内的完整子集），以及数据集B（数据集A删除“冷门话题”后得到）上进行实验，对原系统聚类得到的消息类再进行二次聚类。数据集相关信息和聚类评价指标请参考章节3.3，新系统二次聚类时使用的相似度度量请参考章节3.2，新系统采用的聚类方法请参考章节3.1。其中关于聚类评价指标和聚类方法的实现，NMI的计算、K均值聚类法、谱聚类法使用了Wen-Yen Chen等人实现的版本[15]，基于核函数的K均值聚类法使用了Mo Chen实现的版本[[1]](#footnote-1)（使用高斯核函数），而层级聚类使用了Matlab的实现版本[[2]](#footnote-2)（算法使用的连结方式共5种，请参考章节3.1）。实验结果如下：

表 3.1 数据集A的聚类结果（指标：NMI）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method |  |  |  |  |  |  |  |  |  | Avg |
| Spectral | 0.9313 | 0.8756 | 0.9107 | 0.9321 | 0.9363 | 0.9288 | 0.9384 | 0.9265 | 0.9253 | 0.9227 |
| H-Sgl | 0.9282 | 0.8694 | 0.8876 | 0.9146 | 0.9485 | 0.8480 | 0.9623 | 0.9008 | 0.8809 | 0.9044 |
| H-Cpl | 0.9304 | 0.8858 | 0.9201 | 0.9476 | 0.9552 | 0.9097 | 0.9625 | 0.9231 | 0.9323 | 0.9296 |
| H-Avg | 0.9301 | 0.8817 | 0.9189 | 0.9459 | 0.9573 | 0.9075 | **0.9658** | 0.9213 | 0.9222 | 0.9278 |
| H-Wtd | 0.9301 | 0.8817 | 0.9185 | 0.9461 | 0.9571 | 0.9126 | **0.9658** | 0.9213 | 0.9304 | 0.9292 |
| H-Wrd | 0.9309 | 0.8898 | 0.9255 | 0.9474 | 0.9544 | 0.9195 | 0.9617 | 0.9236 | 0.9294 | **0.9313** |
| Kmeans | 0.9294 | 0.5159 | 0.7034 | 0.9296 | 0.9349 | 0.8903 | 0.9380 | 0.9216 | 0.9164 | 0.8532 |
| Kn-Km | 0.8491 | 0.5116 | 0.6928 | 0.9012 | 0.9056 | 0.8966 | 0.9191 | 0.8996 | 0.8882 | 0.8293 |
| Avg | 0.9199 | 0.7889 | 0.8596 | 0.9330 | 0.9436 | 0.9016 | **0.9517** | 0.9172 | 0.9156 | 0.9034 |

表 3.2 数据集B的聚类结果（指标：NMI）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method |  |  |  |  |  |  |  |  |  | Avg |
| Spectral | 0.7463 | 0.6069 | 0.6609 | 0.6698 | 0.6763 | 0.6961 | 0.8662 | 0.7091 | 0.7131 | 0.7049 |
| H-Sgl | 0.7347 | 0.5478 | 0.5217 | 0.6203 | 0.7666 | 0.5007 | 0.8723 | 0.5038 | 0.4959 | 0.6182 |
| H-Cpl | 0.7420 | 0.5885 | 0.5578 | 0.8024 | 0.8677 | 0.5830 | 0.8957 | 0.6966 | 0.7300 | 0.7181 |
| H-Avg | 0.7445 | 0.5482 | 0.5342 | 0.8130 | 0.8905 | 0.4769 | **0.9299** | 0.6951 | 0.6960 | 0.7031 |
| H-Wtd | 0.7428 | 0.5444 | 0.5313 | 0.8131 | 0.8844 | 0.4731 | 0.9241 | 0.6860 | 0.7027 | 0.7002 |
| H-Wrd | 0.7452 | 0.6175 | 0.6410 | 0.8132 | 0.8644 | 0.6572 | 0.8907 | 0.7289 | 0.7429 | **0.7445** |
| Kmeans | 0.7451 | 0.4780 | 0.5545 | 0.7488 | 0.8118 | 0.6088 | 0.8436 | 0.7039 | 0.7305 | 0.6916 |
| Kn-Km | 0.7168 | 0.5581 | 0.6467 | 0.6692 | 0.7003 | 0.6932 | 0.7990 | 0.6951 | 0.7141 | 0.6880 |
| Avg | 0.7396 | 0.5611 | 0.5810 | 0.7437 | 0.8077 | 0.5861 | **0.8776** | 0.6773 | 0.6906 | 0.6961 |

对实验结果的表头进行一定说明：最左边一列是聚类方法（Method），分别有谱聚类（Spectral），层级聚类（简单连结H-Sgl，完全连结H-Cpl，平均连结H-Avg，加权连结H-Wtd，沃德方法H-Wrd），K均值聚类（Kmeans），基于核函数的K均值聚类（Kn-Km），以及列均值（Avg）。最上方一列除Method的表头均为相似度矩阵的类型，分别是时间相似度（），话题标签相似度（），命名实体相似度（），单词集合的Jaccard相似度（），词频-逆文档频率相似度（），将以上6种相似度进行加权平均的加权相似度（），两种用于对照实验的基准相似度（，），以及行均值（Avg）。表格右下角是除行列均值外所有表格数据的总均值。其中行、列均值以及所有数据的最大值都已用粗体标识出来了。

实验中两种用于对照的基准相似度，它们都只考虑了消息之间文本内容的相似性，采用的度量是语义相似度，使用的工具包为SEMILAR[20]，选用了包中计算句子间相似度的CM法[21]（）和贪心法[22]（），这两种算法的核心都是基于单词网[23]（WordNet）中单词与单词的相似度度量。

实验中将所有方法的聚类类别数量（如K均值法中的K值）都设置为与真实话题数量一致（数据集A中为686类，数据集B中为67类）。

表 3.1为数据集A的二次聚类结果。按行看，聚类结果最好的是使用沃德方法的层级聚类法（H-Wrd），NMI均值为0.9313，实际上除了简单连结外的所有层级聚类法表现都非常好，NMI均值都接近0.93；而表现第二好的是谱聚类法（NMI均值为0.9227），然后是K均值法，最后是基于核函数的K均值法。按列来看，前六列为章节3.2中提出推荐的6种相似度，表现最好的是词频-逆文档频率相似度（NMI均值0.9436），紧接其后的是单词集合的杰卡德相似度、时间相似度以及用户互动相似度，而命名实体相似度的聚类结果倒数第二，话题标签相似度的结果最差（NMI均值0.7889）。而综合考虑这6种相似度的加权相似度则是所有相似度矩阵中聚类表现最好的（NMI均值达到0.9517），在采用平均连结和加权连结的层级聚类时，更是达到了所有实验结果的最大值（NMI为0.9658）。至于两种对照的基准相似度，它们的NMI值都在0.915左右，本文采用的6种相似度中比它们表现好的相似度为词频-逆文档频率相似度单词、集合的杰卡德相似度以及时间相似度，而用户互动相似度虽不如基准但与其差不多，但命名实体相似度和话题标签相似度则远不如基准方法。

与表 3.1相比，表 3.2总体的NMI指标都要低很多，原因是数据集A中的真实类别数目大、每个类别中含有的元素数目少，这容易造成NMI指标计算的结果偏大；而数据集B经过了“冷门话题”的剔除，仅剩下67类共214个元素，其真实类别数目小、每个类别中含有的元素数目多，换言之将其正确聚类的难度大，因此总体NMI偏小，此属于正常现象。

下面来观察数据集B的二次聚类结果（表 3.2）。按行来看，其中表现最好的聚类算法仍是使用沃德方法的层级聚类法（平均NMI为0.7445），同数据集A上的结果类似，各种层级聚类法（除了使用简单连结）的聚类效果都很好（平均NMI都在0.7以上），而谱聚类法与其基本相当（平均NMI为0.7049），而普通的和基于核函数的K均值法表现仍是最差（平均NMI分别为0.6916和0.6880），但在数据集B上所有算法平均表现相差不大。按列来看，6种相似度矩阵的表现排名与数据集A一致，词频-逆文档频率相似度（NMI均值0.8077）远远超出其它5种相似度，接着是单词集合的杰卡德相似度和时间相似度（NMI均值分别为0.7437和0.7369），表现最差是用户互动相似度、命名实体相似度和话题标签相似度（NMI均值都在0.58左右），使用它们单独进行聚类的效果非常不好。同数据集A的情况一样，综合考虑这6种相似度的加权相似度的聚类表现最好，其NMI均值达到0.8776，比单独表现最好的词频-逆文档频率相似度的NMI均值（0.8077）高出8.7%；而在使用加权相似度时，如果采用平均连结的层级聚类法所得到的聚类结果是所有实验结果中最好的（NMI为0.9299），远远超出其它聚类结果很多。最后，两种基准方法的NMI均在0.68左右，与数据集A情况相同，6种相似度中排名前三的在基准之上，排名后三的在基准之下。

通过以上的数据观察，我们可以发现无论是在数据集A还是数据集B上进行二次聚类，其实验结果都有一些共同特点，下面进行实验分析。

在两个数据集中，表现最好的聚类算法都是层级聚类法。这其实是符合我们预期的（参见章节3.1最后），由于社交网络的消息通常是一个用户看到其它用户的消息后发布的，所以讨论同一话题的消息集合中的大部分消息，都与其它至少一条消息高度相似，因此理论上基于贪心思想的层级聚类法，对社交网络中的话题聚类问题有着非常突出的优势。而在两个数据集中，谱聚类法的聚类结果也非常好，原因是它能抽取相似度矩阵中的主要特征，去除噪声后再进行聚类将有助于结果的提升。由于社交网络消息比较杂乱，特征向量分布不均匀，因此K均值聚类法“球形聚类”的特点不适用于这种场景，造成其聚类结果较差。然而基于核函数的K均值分类法表现更糟糕，这表示在这两个数据集上特征向量原本就比较线性可分，不需要核函数进行映射，又或者是核函数选择不当，应选择高斯函数以外的其它核函数。

在两个数据集中，表现最好的相似度矩阵是词频-逆文档频率的余弦相似度。这也是符合我们预期的（参见章节3.2），因为此相似度能将定义“关键词”的任务自动化，将较为独特的单词权重调高，从而更准确地判定两文本的相似度。与它相比，单词集合的杰卡德相似度则没有考虑“关键词”的概念，仅仅是计算两文本单词的重叠率，虽然也是很有效的文本相似度判定方法，但其表现比词频-逆文档频率落后一大截。但在章节3.2也提到过，单词集合的杰卡德相似度非常适合社交网络这种消息内容重复率很高的数据集，因此其表现可以达到第二。时间相似度的聚类表现则是与杰卡德相似度非常接近，这说明即便不参考文本内容，只考虑时间间隔的因素也是能较好地完成社交网络中的话题聚类任务，因为相关话题的消息容易在短时间内集中爆发（特别是热门话题），而基于指数衰减函数的时间相似度能突出这种时间上的相似性，所以它也很适合社交网络的话题聚类。然而用户互动相似度、命名实体相似度以及话题标签相似度的表现在两数据集中都很差，甚至远不如基准方法的相似度，这是因为用户互动、命名实体和话题标签并不是每条消息都必然拥有的元素，它们存在时能成为话题聚类的强线索，但其稀疏性决定了它们不能被单独用作相似度进行聚类的事实；但是，将它们作为辅助因素加入到整体的相似度矩阵中却可以一定程度地增强话题聚类效果。

下面来看看综合考虑了6种相似度的加权平均相似度，无论它套用哪种聚类方法，其聚类效果都是卓越的，比所有其它类型的相似度矩阵都高出很多。这是非常自然的，因为它用权重配比的方式将时间、用户互动以及文本内容等多方面因素结合起来，这样既能让效果比较好的相似度作为主导，又能同时考虑其它稀疏的相似度矩阵，用它们辅助增强聚类。但唯一的问题是，加权相似度的权重配比应该如何决定？在实验中，本文采取了网格搜索的方式遍历配比参数，具体是以0.1为单位，将总权重1.0不均等地分给6种相似度，并遍历所有分配方式，找到聚类结果最好的配比。但是这样进行迭代取最优的参数优化方法，有可能会引起过拟合的问题，因为适用于一种聚类方法的配比不一定适用于另一种，而且适用于测试数据集的配比也不一定适用于其它数据集。那么如果系统要对一个新来的数据集进行聚类，我们应该使用怎样的权重配比呢？

针对这种情景的过拟合问题，一般有两种解决思路：要么通过先验知识进行人为地分配，例如我们已经知道词频-逆文档频率相似度效果很好，而话题标签相似度效果较差，那么我们可以主观地给前者分配较高的权重，后者分配较低的权重，但具体分配多少就要靠分配者的经验与直觉，这样分配出来的权重不一定能保证是最优的，但有经验的分配者一般能分配出较好的比重；而另一种则是基于统计学的方法，通过更换大量的测试数据集，或者更换不同的聚类方法，将大量的最优配比进行平均，获得一个较为“普适”的权重配比，这样的配比同样不能保证在新数据集上可以获得最优的结果，但它是很多个最优结果的平均，理论上在大部分数据集中其聚类结果都不会太差，就像“万金油”一样。

本文将通过类似后一种的解决方案，给出一个推荐的权重配比：将数据集A和数据集B中所有聚类方法网格搜索出来的最优配比进行平均，获得一个“普适”的权重（仅限于本实验），两个数据集下各聚类方法获得最优聚类效果（最高NMI）时，加权相似度的权重配比表如下：

表 3.3 各聚类方法在两数据集下最优的权重配比

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Mtd-Ds |  |  |  |  |  |  | Total |
| Spetral-A | 0.5 | 0.0 | 0.0 | 0.0 | 0.4 | 0.1 | 1.0 |
| H-Sgl-A | 0.1 | 0.1 | 0.0 | 0.0 | 0.8 | 0.0 | 1.0 |
| H-Cpl-A | 0.1 | 0.3 | 0.0 | 0.0 | 0.6 | 0.0 | 1.0 |
| H-Avg-A | 0.1 | 0.1 | 0.0 | 0.0 | 0.8 | 0.0 | 1.0 |
| H-Wtd-A | 0.1 | 0.1 | 0.0 | 0.0 | 0.8 | 0.0 | 1.0 |
| H-Wrd-A | 0.1 | 0.2 | 0.0 | 0.0 | 0.7 | 0.0 | 1.0 |
| Kmeans-A | 0.0 | 0.4 | 0.0 | 0.0 | 0.5 | 0.1 | 1.0 |
| Kn-Km-A | 0.1 | 0.2 | 0.0 | 0.4 | 0.3 | 0.0 | 1.0 |
| Spetral-B | 0.1 | 0.1 | 0.1 | 0.2 | 0.5 | 0.0 | 1.0 |
| H-Sgl-B | 0.1 | 0.1 | 0.1 | 0.4 | 0.3 | 0.0 | 1.0 |
| H-Cpl-B | 0.1 | 0.2 | 0.0 | 0.0 | 0.7 | 0.0 | 1.0 |
| H-Avg-B | 0.1 | 0.0 | 0.0 | 0.2 | 0.7 | 0.0 | 1.0 |
| H-Wtd-B | 0.1 | 0.2 | 0.1 | 0.0 | 0.6 | 0.0 | 1.0 |
| H-Wrd-B | 0.1 | 0.1 | 0.0 | 0.3 | 0.5 | 0.0 | 1.0 |
| Kmeans-B | 0.1 | 0.0 | 0.0 | 0.1 | 0.8 | 0.0 | 1.0 |
| Kn-Km-B | 0.2 | 0.3 | 0.1 | 0.0 | 0.4 | 0.0 | 1.0 |
| Avg | 0.125 | 0.15 | 0.025 | 0.1 | 0.5875 | 0.0125 | 1.0 |

表格最左边一列是“聚类方法-数据集”（Mtd-Ds），聚类方法命名同之前的实验，两数据集用A、B表示。对所有最优配比取平均后得到的“普适”配比在最后一行，可以看到此配比是基本符合我们之前的分析的：最有效的词频-逆文档频率相似度获得了最高权重，而单独使用时效果较好的时间相似度与单词杰卡德相似度也获得了较大比例，而单独使用时表现糟糕的命名实体相似度和用户互动相似度则得到了很低的分配比重；但比较违反直觉的是，独立使用时表现不好的话题标签相似度，却获得了第二大的比重，当我们再仔细观察后发现基本每个最优配比都会给话题标签相似度赋予一定的权重，这说明话题标签虽然单独使用时无法很好地聚类，但在结合使用时却是一个比较重要的辅助因素，经常能加强整体的聚类效果；另外单词杰卡德相似度获得的比重并没有预期的高，原因很可能是词频-逆文档频率相似度已经很大程度地包含了杰卡德相似度的作用，而它本身由于没有考虑“关键词”所以有一定的噪音，于是在结合考虑6种相似度时，它有时就比较冗余，因此平均后仅能得到中等的权重。

本文使用这个平均的权重配比，又重新在两个数据集下评测各聚类方法的聚类表现，得到以下实验结果表格：

表 3.4 使用统计平均配比的加权相似度的聚类结果（指标：NMI）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | Spectral | H-Sgl | H-Cpl | H-Avg | H-Wtd | H-Wrd | Kmeans | Kn-Km | Avg |
| A | 0.9328 | 0.9591 | 0.9619 | **0.9660** | 0.9645 | 0.9615 | 0.9342 | 0.9061 | 0.9482 |
| B | 0.8394 | 0.8368 | 0.8752 | 0.9009 | **0.9035** | 0.8795 | 0.7981 | 0.7508 | 0.8480 |

通过对比表 3.4和表 3.1、表 3.2，我们发现使用统计平均配比的加权相似度矩阵，在两数据集中用各种聚类方法，在大多数情况下都表现很好，跟网格搜索的最优结果持平或相差不远（NMI均值0.9517 vs 0.9482；0.8776 vs 0.8480），虽然总体仍不如网格搜索；只有在少数情况下会出现较大差距，如数据集B上的K均值和基于核函数的K均值算法（NMI值0.8436 vs 0.7981；0.7990 vs 0.7508）；而在极少数时候，统计平均的配比反而网格搜索的最优配比结果更好，如数据集上A使用平均连结的层级聚类（NMI值0.9658 vs 0.9660），这是因为网格搜索的“最优”结果其实并不是真正的最优，只是在网格搜索过的参数集下是最优的，像本实验是以0.1为单位将总权重1.0分配给6种相似度矩阵，实际上只遍历了3003种权重配比方案，这就漏掉了如统计平均配比这样的更为“精细”的配比，因此统计平均配比比网格搜索最优配比的聚类效果更好是完全有可能的（虽然这并不常见）。

因此结论是推荐的统计平均配比，一定程度可以适用于不同数据集下的各聚类算法中。若想获得更加稳定的加权矩阵配比，则应引入更多数据集。

在本实验的最后，我们来回顾一下我们最初提出的问题有没有得到解决。我们之所以要在原系统筛选聚集出来的消息类的基础上，再做一次二次聚类，是因为原系统筛选出来的消息类话题重复率过高。例如在删除“冷门话题”的数据集B中，共涉及8,373条谣言候选消息，原系统将它们聚成的候选消息类共有214个（也即平均每类39条消息），经人工标注后得知真正不同的话题仅有67类（也即平均每类125条消息），计算得话题重复率高达3.2倍。

那么我们通过二次聚类后得到的新消息类又有几类呢？有没有将重复的话题聚集在一起？回答这些问题前我们首先要明白，实际上二次聚类后得到的新消息类的总类数是算法预先定义好的：例如K均值法和谱聚类法都要在算法启动前预定义聚类中心的数目，也就是聚类后的类数（K）；再如层级聚类法，既可以直接设定类数，也可以通过设定一个停止聚类的距离阈值来间接控制类数，但是一旦这个参数定了，相同的数据聚类而成的类数也是确定的。所以实际上我们想得到几类就能得到几类，于是问题就来了：我们怎么知道最佳的预定义类数是多少呢？又怎么能确定新系统的确能降低话题重复率，并正确聚类？有的读者可能会想，我们通过遍历参数的方法，求出最高的NMI对应的类数不就是最佳类数了吗？这是个有趣的想法，但可惜的是实际问题中我们并不能这么做。因为目前我们之所以能求NMI，是因为我们将一个小数据集上的消息进行了聚类标记，而在实际应用中，我们并没有聚类标签，也不可能给任何稍大的数据集进行标记（实际上我们正是不想去标记才研究聚类算法的）。所以以上遍历参数找最优指标的方法不适用于系统求解真实问题，在真实场景下确定最佳类数，一般的做法只有凭经验或先验知识，人为地尝试几组参数，通过小范围地查看聚类结果来判断参数是否合理。

虽然在真实场景下最佳类数很难确定，但是我们仍可以评估新系统二次聚类后的效果，以及它能否在压缩话题重复率的同时保持良好的聚类效果（在预定义类数接近真实值时，聚类效果不会太差）。用到的方法正是上文提到的遍历类数找最优NMI：我们以加权相似度为例，取统计平均权重配比（表 3.3最后一行），聚类方法选用效果最好的层级聚类，尝试将类数定为不同的值来运行算法，计算其在数据集B上的NMI值。实验后获得以下表格：

表 3.5 不同的预定义类数K的聚类效果（指标：NMI）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| K | K / Ktrue | H-Sgl | H-Cpl | H-Avg | H-Wtd | H-Wrd |
| 30 | 0.447 | 0.5774 | 0.8016 | 0.8312 | 0.8164 | 0.8084 |
| 40 | 0.597 | 0.6601 | 0.8384 | 0.8489 | 0.8502 | 0.8438 |
| 50 | 0.746 | 0.7882 | 0.8518 | 0.8686 | 0.8707 | 0.8673 |
| 60 | 0.895 | 0.8149 | 0.8669 | 0.8977 | 0.8948 | 0.8749 |
| 67 | 1.000 | 0.8368 | 0.8752 | 0.9009 | 0.9035 | 0.8795 |
| 70 | 1.044 | 0.8489 | 0.8842 | 0.9094 | 0.9073 | 0.8832 |
| 80 | 1.194 | 0.8771 | 0.8930 | 0.9139 | 0.9108 | 0.8896 |
| 90 | 1.343 | 0.8906 | 0.9045 | 0.9160 | 0.9177 | 0.8994 |
| 95 | 1.417 | 0.9083 | 0.9076 | 0.9180 | **0.9178** | 0.9004 |
| 100 | 1.492 | 0.9126 | **0.9095** | 0.9205 | 0.9163 | 0.8986 |
| 105 | 1.567 | 0.9115 | 0.9074 | 0.9162 | 0.9133 | **0.9036** |
| 110 | 1.641 | 0.9102 | 0.9028 | **0.9218** | 0.9126 | 0.9035 |
| 115 | 1.716 | 0.9101 | 0.9012 | 0.9139 | 0.9139 | 0.9030 |
| 120 | 1.791 | **0.9189** | 0.9007 | 0.9121 | 0.9112 | 0.9014 |
| 125 | 1.865 | 0.9107 | 0.9004 | 0.9053 | 0.9066 | 0.8983 |
| 130 | 1.940 | 0.9077 | 0.8987 | 0.9033 | 0.9004 | 0.8978 |

表 3.5最左边的一列（K）是算法预定义类数，第二列（K / Ktrue）是预定义类数与真实类数的比值，后面几列是不同连结方式的层级聚类法在数据集B上的聚类效果。从表格可以观察到，在预定义类数等于真实值（67）的时候，各聚类算法的NMI值都还不错，与最佳指标相差都不大且均在0.83以上；当K值在真实值左右波动时（6720）时，NMI指标也都属于可接受的范围或以上。因此可以认为以加权相似度矩阵为内核的层级聚类算法，在压缩话题重复率的同时能够保持良好的聚类效果。再来观察一下当算法获得最优聚类结果时，系统能将话题重复率压缩到多少。表 3.5已经将各连结方式层级聚类法的最高NMI值以粗体标出，对应的话题重复率分别是1.791、1.492、1.641、1.417、1.567，范围在1.4到1.8之间，也就是说从NMI的角度，系统在聚类效果最好时能将话题重复率平均降到1.6左右，而原系统筛选聚类后得出的消息类，其话题重复率是3.2，也就是说新系统将原系统的话题重复率整整压缩了一倍。即使新系统的话题重复率1.6相比理想的1.0仍有一段距离，但可以说本文的改进显著改善了原谣言检测系统话题重复率过高的问题。

最后，我们来通过一些例子来说明新系统的确能将讨论同一话题的消息类聚集在一起，表格如下：

表 3.6 消息举例：新老系统聚类对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ltrue | T# | Lold | Representative Tweet Text | T# | Lnew |
| 1 | 26 | 1 | RT @PunchingYou: 4chan is trying to spread some shit rumor about ebola. If you see the hashtag "#ebolaindoritos", it's horseshit. http://t.co/PSriKtBaoX | 15 | **1** |
| 2 | I went out to get some General Tso's, 4chan. Still watching your crappy rumor. #ebolaindoritos | 1 | **1** |
| 3 | RT @Shaftgodd: "@MiddleNamesSeth: Ebola has contaminated my favorite snack. #EbolaInDoritos http://t.co/BnqwJbJb6R"IS THIS TRUE IT CANT BE | 5 | **1** |
| 4 | Also, a fine round of applause for rumormongering goes to these 4channers. #ebolaindoritos http://t.co/r0u6eCo3K5 | 1 | **1** |
| 5 | If you're curious who to thank for this shit rumor, try these folks. A friendly 'fuck you' to #ebolaindoritos http://t.co/XbLXQfYKFq | 1 | **1** |
| 6 | RT @wafflesatnoon: Don't believe the rumor that Ebola is found in Doritos. That is a hoax by 4chan, debunked by @Fritolay . http://t.co/4hC | 3 | 2 |
| 2 | 12 | 7 | da fuq?! RT @PrettyCrazy3 starledger: A number of Trenton residents are being monitored for Ebola symptoms, health officer says?what? | 9 | **3** |
| 8 | what?!?! they think people in Trenton have Ebola?! @MinaSayWhat | 1 | **3** |
| 9 | People being monitored. Get your Camron Ebola masks! RT @JerseyImperator: what?!?! they think people in Trenton have Ebola?! @MinaSayWhat | 2 | **3** |
| 3 | 883 | 10 | RT @RealBlindPirate: Really? #Obama asks for $6.2 billion freakin dollars to fight #Ebola? Really? REALLY?!? Un-freakin BELIEVABLE! | 3 | **4** |
| 11 | What? MyFoxTampaBay: Pres. Obama seeking $6.2 billion to confront Ebola in West Africa and prevent spread in US. (@AP)? | 563 | **4** |
| 12 | RT SavienPayne: RT callmeMIMIbaby: What?! RT AP: BREAKING: Obama seeking $6.2 billion to confront Ebola in West Africa and prevent spread... | 284 | **4** |
| 13 | Obama's Ebola request:$6.2B($2B USAID, $2.4BHHS,1.5B contingency fund?)Really? Where is additional funding for first responders & troops. | 1 | 5 |
| 14 | Pres.O asked congress for $6.2B to fight ebola in WestAfrica? Really?It costs ZERO to DENY ENTRY2 ppl who've bn in WA in last 21 days | 1 | 6 |
| 15 | #Obama wan $6 bil. 4 #ebola. But its not going all 2 ebola? What? #Vacation pocket money? | 1 | 7 |

表 3.6的第一列为真实的话题标签（Ltrue），第二列为每个真实话题的消息数量（T#）；第三列到第五列都是原系统筛选聚类生成的消息类的信息，第三列是原消息类标签（Lold），第四列是原消息类中“代表消息”的文本内容（Representative Tweet Text）第五列是元消息类中含有的消息总数（T#）；而表格第六列是新系统将原消息类进行二次聚类后得到的新消息类标签（Lnew）。此例子取自表 3.1中H-Avg行、simw列的实验结果，是新系统用网格搜索在数据集A上得到的最佳聚类效果（NMI值最高）。

观察表 3.6给出的例子，真实话题有3个，原候选消息类却有15个，而二次聚类后新候选消息类仅7个，话题重复率被压缩到一半以下。其中第一个话题是关于网站4chan上散播的一条谣言，其称Doritos公司的零食含有埃博拉病毒。观察其中6个原消息类的代表消息文本，发现其单词重叠率其实不高，有时重复的关键词还被不同的词性割裂了（如4chan和4channers），但前5则消息都有且仅有一个共同的话题标签“#ebolaindoritos”，这就会导致它们的话题标签相似度很高（为1.0），而且它们的发布时间都在11月3日或4日，这导致它们也有很高的时间相似度；因此即便这5个消息类的文本重叠率不高，但却能被基于加权相似度的聚类方法成功聚为一类；而第6个原消息类仍没有被新系统正确聚类，原因是此消息类文本中不含话题标签“#ebolaindoritos”，这降低了它与其它5个消息类的加权相似度。

而第二个话题是谈论一则成特伦顿（Trenton）发现埃博拉病患的谣言，这个话题含有的3个原消息类，都被新系统成功地聚在一起。这是因为作为重叠关键词的“Trenton”的逆文档频率很高、独特性强，所以导致它们的词频-逆文档频率相似度很高，另外“Trenton”作为地名也是命名实体，而且后两则消息有相同的提及用户“@MinaSayWhat”，这两种个因素也会辅助加权相似度进行聚类。

第三个话题谈论的是一则非谣言的新闻（换言之这是一个识别错误的谣言候选话题），新闻谈论的是美国总统奥巴马向国会申请62亿美元来抗击埃博拉。新系统对这个话题的聚类效果不太理想，只将前三类成功聚起来，而后三类却分别自成一类。前三类之所以能聚集起来是因为他们都有重叠词“Obama”和“$6.2”，而且“$6.2”的逆文档频率较高；而后三类聚类失败则是因为有的消息的“Obama”写成了“Pres O”，而有的消息的“$6.2”写成“$6.2B”或“$6 bil”，这些因素会显著降低词频-逆文档频率相似度和单词集合的杰卡德相似度，另外后三类并没有共同的话题标签、命名实体，消息文本中也没有提及符号，故辅助的因素都无法其作用，导致了使用加权相似度的新系统也无法将它们正确聚类。这也是正是社交网络话题聚类所面临的挑战：消息内容过于杂乱，相似的特征因素稀疏，导致聚类难度大大增加。

虽然不尽完善，但从表 3.6的例子可以看出，改进后的新系统能够将部分话题重复的消息类成功聚类，大大降低了系统筛选话题的重复率，进而显著降低后期对谣言候选话题进行复核的人工成本。

谣言系统中关于降低话题重复率的部分就到此为止了。除了话题重复率过高，原谣言检测系统还面临着一个问题：检测准确率不高。下两章将分别从特征选择的角度和监督学习分类器的角度，尝试对原系统进行改进。

# 特征选择

## 特征选择简介

任何监督学习技术都需要输入一定的特征集合，通常输入的特征越多样化，特征越能区分样本对象则分类准确率越高。当训练样本量足够大，而训练时间又没有限制的时候，输入越多种类的特征越容易得到正确的分类结果。但现实中因为标记数据成本高，训练样本通常很少同时又存在很多噪声样本，这种情况下输入过多的的特征反而很可能导致训练过拟合（overfitting）。

特征选择技术的目的正是解决由过多特征引起的过拟合问题，其基本任务是：给定一个特征集合，找出其中那些对解决分类问题有较大帮助的特征，同时尽量缩小特征的数目，将冗余特征从集合中去除，最终选出一个对分类问题有效且紧凑的特征子集。通常此类研究将特征定义为三种类型：关于类别的强相关特征、弱相关特征和无关特征[24, 25]。选择算法通常致力于选出所有强相关特征，以及数量较少、效用较高的部分弱相关特征，同时将无关特征排除。

通常特征选择技术分为三大类[24]：过滤器（filter）、包装器（wrapper）以及嵌入式（embedded）。考虑到嵌入式方法通常是将特征选择过程嵌入特定分类器的学习过程中，是一类针对特定分类器进行特定设计的选择技术，一般不具有普遍性，因此本文系统仅采用了前两种特征选择技术。以下两个小节将对系统用到的过滤器和包装器特征选择进行概述。

## 过滤器

过滤器（filter）是最早的一类特征选择技术，其命名思想是将特征集合中的相关特征过滤出来，得到有效且不冗余的特征子集。这一类技术的特点是特征选择过程中没有分类器的参与，仅仅依靠样本特征数据和样本标签，通过相关性分析挑选特征。通常使用到的相关性指标有皮尔曼相关系数[28]（Pearman correlation）、斯皮尔曼相关系数[29]（Spearman correlation）以及互信息[30]（mutual information），本系统的实现采用了前两种相关系数，以及互信息的改进版本归一化互信息[19]（normalized mutual information）。相关性指标只是一个评估函数，事实上它完全可以被更换为任何有效的度量，而传统的过滤器技术则是基于评估函数的一套特征选择算法。本节将介绍本系统采用的两种过滤器技术。

如何挑选与类标签相关的特征？一种比较朴素的想法是先将所有特征和类标签合理地数值化，然后对每个特征变量利用评估函数计算它与类标签变量的相关度，最后选出相关度排名前N的那些特征就能形成一个大小为N的相关特征子集。这个想法非常直观，而事实上早起的特征选择算法也是这么做的。但是这种做法存在许多问题，其中一个是这样挑选出来的特征集不能防止冗余特征的出现。假设某特征对类标是弱相关的，虽然它也与标签变量相关度较高，但这种关联已经被另几个强相关的特征涵盖了，那么它很可能是一个冗余特征，但也会被算法所选择。为了防止这种情况的出现，研究者们提出了一些增强型的过滤器算法。

其中一种比较典型的增强型过滤器算法，是在评估函数中引入与其它特征的相关性评估。如Hanchuan Peng等人的研究[26]，其将评估函数定义为以下形式：

公式中E是评估函数，S是已经选择的特征集合，I是某种相关度度量（在原文中它是互信息），c是类标签变量，xj是未被挑选的特征集合中的某个特征变量，xi是S中的特征变量（算法已经选择的特征）。此过滤器算法是个贪心算法：一开始S为空集，公式中的第二项被忽略，只选择出相关度最高的特征加入S；然后开始进行反复迭代选取E值最高的特征加入S，直到满足某条件停止迭代。

公式第一项为常规的相关度度量，符号为正是希望选择到与类标签相关度高的特征；公式第二项则评估了未选择的某特征与已选择各个特征的相关度并求平均值，这项是负数是希望选出尽量独立于已选择特征集的、不冗余的特征；如果将第二项去除，那么算法就退化成挑选相关度最高的前N个特征。

此评估函数的设计综合考虑了相关性和冗余性两方面的因素，整个选择的算法流程简单且时间复杂度低，因此被广泛用于快速挑选出有用特征。本文也采用了此算法来进行特征选择，在具体实现中相关性度量I选择尝试了归一化互信息、皮尔曼相关系数的绝对值以及斯皮尔曼相关系数的绝对值三种度量。

虽然此算法速度快，也考虑了两方面的因素，但实际上它每次只考虑了变量两两间的相关性，而实际上好的特征集合不一定是每个特征都与类标签相关性很高，也不一定特征之间相关性都很弱，更多时候可能是多个特征的组合与类标签相关度高，特征之间有一定的相关性却各有独特的性质进行互补。由于此算法没有考虑特征集合整体的作用，因此选出来的特征集合质量通常不是很高。

以上的是基于相关性度量的过滤器算法，其选择质量的高低依赖于合理的相关性度量，于是大量研究者致力于提出更有效的评估函数。但此外也有部分研究者索性不往这方面深入研究，他们将相关性的分析任务交给大量样本实例来完成。

Kenji Kira等人的缓和算法[27]（relief algorithm）就是基于样本实例的。他们算法的核心是先将样本投影到特征空间，然后对每一个样本实例，找到在特征空间中与它距离最近且标签相同的另一个样例，另外也找出特征空间中与它距离最近且标签不同的另一个样例，前者称为该样例的近邻命中样例（near-hit），后者称为该样例的近邻失误样例（near-miss）。然后对每个特征进行如下的权重迭代更新：

其中wi是第i个特征的权重，而xi是样本第i个特征值，near-hiti是近邻命中样例第i个特征值，near-missi是近邻失误样例的第i个特征值，diff是特征值的距离函数。算法思想就是通过近邻样例与样例间的特征差异，评估每类特征对标签的区分度，例如第二项是负数，表示如果样例的某个特征与近邻命中样例的特征差距很大，则此特征不适合用于分类，反之如果差距很小证明同标签的两个近邻样本都有相同的特征，一定程度说明了此特征可用于分类；第三项也是基于同样的思想，如果两个近邻却类别不相同的样例的某个特征有很大差距，那么此特征会有可能很适合于分类。最后通过遍历大量样本，算法不断更新每个特征的权重，最后挑选出前N个权重最高的样例，作为选择的特征子集。

该算法并没有涉及具体的相关度度量，而是依赖样本和其近邻来告诉我们哪些特征对区分类别有效，哪些特征无效，一般而言越丰富多样的样本能选择出越高质量的特征子集。算法流程很简单，虽然时间复杂度与参考的样本数量有关，但现实中通常训练样本不多因此速度不慢。本系统也采用了这种算法尝试进行特征选择。

如上所述，该算法的思想和权重迭代公式的设计有一定的合理性，但实际应用当中也存在一定问题：第一，算法非常依赖样本丰富程度和多样性，但现实中这两点常常是缺失的；第二，算法只考虑了特征对近邻的区分度，其思路过于微观，选择出来的特征不保证对样本有宏观区分度，且容易引起类似过拟合的问题；第三，不同的特征由于分布、性质不一样，距离函数度量难以统一，这给最后通过比较权重来挑选特征带来很大的麻烦；第四，该算法依然只是考虑了单个特征对分类的影响，如果单个特征无法区分类别，只有在多特征组合时才能有效区分（例如异或运算），那么此算法无法选择出有效的特征子集。

以上的第四个问题是很多过滤器算法面临的挑战，因为考虑特征集整体与类别的关联并不容易做到，如果做到了很可能算法就不仅是特征选择技术，或许已经成为一种分类器技术了。如Claire Cardie的研究[31]就利用决策树来进行特征选择，因为决策树这种分类器的训练一定程度上就是特征选择的过程，而随着树高度的增长，不同特征的整体作用也能突显出来。

因此大部分的过滤器算法还是保持简单、高效的特点，用于从大量特征集中快速筛选出较为可靠的特征子集。由于过滤器算法是独立于分类器的：此过程仅涉及特征和类标的数据分析，没有分类器的参与（这也是高速的原因之一），所以选出的特征是“普适优”的。但是不同分类器有各自的特点，“普适优”的特征子集不一定适合各种分类器。因此还有另一类特征选择技术，它们的目的就是找到适合各分类器的特征子集。这类特征选择统称“包装器”。

## 包装器

不同于过滤器技术对分类器的独立性，包装器特征选择技术由各分类器直接参与特征选择，最终目的是选择出适用于特定分类器的一组特征子集（这样的特征子集不保证适用于其它分类器）。区别于嵌入式的特征选择技术，包装器通常是将分类器当成一个“黑盒子”使用，不进入分内器内部进行算法修改，而仅仅利用其输入输出，因此就像是在分类器外面包上了一层外皮，这就是包装器命名的由来。

包装器算法的思想类似于遍历参数的网格搜索，每次给分类器输入一组候选的特征子集让其在有标数据集中进行训练和分类准确度评估；通过不断输入不同的特征子集，最终找到准确度评估最高的那组特征就是选择的结果。虽然算法很像暴力搜索，但实际上它不是也不可能遍历所有可能的特征子集。因为当特征数目为n，其不同的特征子集除去空集一共有(2n-1)个，指数级别的时间复杂度以及分类器通常不短的训练时间，决定了不能用遍历的方式来选择。包装器算法的核心在于通过启发式的搜索不断更新选择的特征集，最终找到局部最优的特征子集。包装器一般分为两个部分：评估器和搜索引擎。

评估器用于评估某特征子集对分类器准确度的正面效益。由于整个算法属于全自动化的，因此评估只能使用有标记的数据集进行，通常做法是将其中一部分划分为训练样本，剩余部分作为测试数据进行评测。评估指标通常有全测试样本的分类准确度（accuracy）、对于某一类的分类准确率（precision）以及对某一类的检测F1度量（F1-score）等（本文也采用了这三种度量），同时为了防止过拟合，经常会使用交叉验证（cross-validation）的方法计算准确率度量的均值、方差等作为最终的评估指标（本文评估器采用了5折交叉验证取均值的方法）。

而搜索引擎的作用则是确定下一次送入评估器评测的是哪组特征子集。如上文所述，由于遍历法不现实，任何搜索引擎都仅能搜索有限的特征子集，得到一个局部最优的解。因此制定一个较优的启发式搜素策略尤为重要，要在尽可能少的迭代次数内搜索到在全局尽量优的局部最优解。Ron Kohavi等人的文章[25]介绍了两种常用的搜索策略：爬山算法和最优优先算法。爬山算法的算法流程是从空集出发，每次向选择特征集中尝试增加一个未选择的特征（或者从全集出发每次删除一个特征），通过评估器找到最优的增加（或删除）策略并与当前特征选择集的评估指标比较，如果比当前特征选择集更优，则可以在新的特征子集的基础上继续下一轮迭代搜索；如果新的最优特征子集不如当前的特征子集则搜索结束，输出当前特征子集。该文章通过实验论证，爬山算法经常过早地落入局部最优而停止搜索，选择出来的特征子集质量不如另一种搜索策略，最优优先算法。

最优优先的搜索策略，其核心思想是维护一个特征子集的集合F，此集合中的特征子集都进行过评估，但还未对其进行过拓展。这里的拓展是指通过一定的操作将特征子集变换成新的特征子集，如上面爬山算法的给特征子集增添一个未选择特征（或删除一个已选择特征）就是一个操作。算法每轮迭代都从F中挑出评估最优的特征子集f进行拓展，然后将拓展后产生的所有未评估过的新特征子集送入评估器进行评测，然后将它们加入F中，同时把f从F中移除，再进行下一轮迭代。在迭代过程中一直维护一个最优的候选特征子集b，每次有新特征子集进行评估，就将新特征子集的评估指标与b的比较，当且仅当新特征子集评估指标比b的大，则将其设置为新的b。预先给定整数p，当b在p轮中当没有变化则搜索结束（也可以设置最大迭代次数来停止算法）。除此之外，该文章还提出了在每次迭代中融合最优操作的方法，加大搜索到全局最优的可能性。本文系统的实现采用了最优优先的搜索策略以及融合操作的优化方法，搜索模式上尝试了从空集出发仅含增添操作的前向搜索，以及从全集出发仅含删除操作的后向搜索，在章节4.4中还将介绍系统采用的第三种搜索模式。

包装器技术通过迭代搜索找到对特定分类器最优的特征子集，没有与分类器无关的相关性分析，感觉上它比过滤器技术更加直接有效。但其实包装器算法也有各种问题：第一，它选择出来的特征理论上仅适用于特定分类器，当有新分类器需要选择特征，算法必须重新运行；第二，由于采用搜索最优的模式，因此即便有交叉验证，包装器的特征选择还是经常会面临过拟合的问题，在有限的标记样本中追求准确率最高的特征组合并不一定适合分类器去给新数据集进行分类；第三，由于包装器基于评估器，免不了每轮迭代都会进行训练和测试评估，而大部分分类器的训练速度不高，再加上每轮迭代都会新产生很多候选特征子集要评估，以及评估进程需要交叉验证的多次训练，因此包装器的选择时间通常很长；第四，对搜索起点缺乏指导，每次搜索都从空集或全集开始（或使用随机搜索种子）经常搜索出来的局部最优解在全局并不优。针对第四点问题，本文提出了一点改进，下一章节将作介绍。

## 以过滤器为指导的浮动式包装器

过滤器的特征选择高效迅速，却不一定能选出适合特定分类器的特征子集；包装器可以针对特定分类器，通过长时间迭代找到局部最优的特征子集，却因为搜索起点不佳而经常无法覆盖全局较优的特征组合。能否将这两种技术的结合在一起，使他们的优势互补呢？

Hanchuan Peng等人的研究[26]尝试将过滤器和包装器结合在一起，他们的做法是先利用过滤器选出一组特征子集，然后将此特征子集作为全集，让包装器在此特征子集中再寻找更优更紧凑的特征子集。他们用理论和实验证明这样的做法能降低包装器的搜索迭代次数，使其以更快的速度收敛。但是此方法的问题在于越快的收敛速度不一定会导致越好的搜索结果，反而更可能过早落入某个全局较差的局部最优当中；另外，那些未被过滤器选中的特征在第一阶段就直接被剔除了，永远没有机会在第二阶段加入候选特征子集，然而过滤器的相关性分析经常并不可靠，仅凭这样一步就否定某个特征并不妥当。

与他们的研究不同，本文想提高的不是收敛速度，而是搜索到全局更优解的可能性。因此本文提出通过过滤器指导搜索起点的包装器方法。

思路并不复杂，第一阶段同样是使用过滤器选择出一组特征子集，但第二阶段稍有不同：不是让包装器在此特征子集的子集集合中进行搜索，而是将此特征子集作为包装器的搜索起点让包装器开始搜索。这么做的灵感来源是过滤器能高速地找到较优的、不偏向任何分类器的特征子集，也就是一个理论上在任何分类器中表现“还不错”的特征组合，这样的特征子集适合作为搜索种子，以它为基础进行增添特征、删除特征等等操作，拓展发掘出更优的特征子集。这样通过不同过滤器的指导，尝试更多的搜索起点，有助于包装器覆盖到全局更优的解。

需要说明的一点是，这种方法每轮的拓展操作包括增添一个未选特征和删除一个已选特征两种简单操作，以及融合几个最优简单操作的复合操作[25]，因此实际上搜索过程就像是在搜索起点周围进行浮动。这种浮动式搜索相比传统包装器的前向搜索和后向搜索，其每次迭代的搜索广度要大得多，这样一定程度提高了搜索的覆盖范围，使分类器不容易过早陷入局部最优；同时也使得那些被剔除过的特征有机会被添加回来，而那些被尝试添加的特征也有机会被剔除，只要这么做能使特征子集的整体评估变得更优。再加上能尝试不同过滤器生成的不同搜索起点，可以说此方法的确有机会提高包装器得到全局更优解的概率。

但这是一个以时间换取表现的方法：更多的操作、更广的搜索宽度、特征增增减减的来回浮动以及尝试更多的搜索起点，这种种因素使得此方法每轮迭代更耗时、迭代收敛速度更慢，导致算法总耗时经常是一个较高的值。

章节5.4将通过实验对不同的特征选择算法（包括过滤器、包装器、以过滤器为指导的浮动包装器）选择出来的特征子集对分类器分类准确度的效益进行全面的评测，证明在原系统中引入特征选择技术有助于系统谣言检测准确率的提高。

## 系统引入的新特征

由于原系统中仅有13类特征，其特征多样性不足会成为阻碍系统提高检测准确率的瓶颈。因此本文在调研了社交网络中谣言检测的相关研究后引入了大量的新特征，将特征扩充成了45类。需要注意的是由于本文系统是对话题进行谣言识别，因此以下这些特征的主体不是消息，而是每一个候选消息类，也就是经过聚类后的一组相似消息的集合，因此很多特征是统计量，或者经过计算均值生成的。完整的特征列表如下：

表 4.1 新系统使用的特征列表

|  |  |  |
| --- | --- | --- |
| 类型 | 特征 | 描述 |
| 原系统特征  （共15类） | 信号消息比例 | 候选消息类中含信号消息的比例 |
| 信号消息平均字符长度 | 候选消息中信号消息字符长度的平均值 |
| 所有消息平均字符长度 | 候选消息中所有消息字符长度的平均值 |
| 信号消息字符长度比 | 比值：信号消息平均字符长度 / 所有消息平均字符长度 |
| 信号消息平均单词长度 | 候选消息中信号消息单词个数的平均值 |
| 所有消息平均单词长度 | 候选消息中所有消息单词个数的平均值 |
| 信号消息单词长度比 | 比值：信号消息平均单词长度 / 所有消息平均单词长度 |
| 信号消息转发比 | 属于转发消息的信号消息占候选消息类中所有信号消息的比例 |
| 所有消息转发比 | 属于转发消息的消息占候选消息类中所有消息的比例 |
| 信号消息平均url数量 | 候选消息类中信号消息url数量的平均值 |
| 所有消息平均url数量 | 候选消息类中所有消息url数量的平均值 |
| 信号消息平均标签数量 | 候选消息类中信号消息hashtag数量的平均值 |
| 所有消息平均标签数量 | 候选消息类中所有消息hashtag数量的平均值 |
| 信号消息平均提及数量 | 候选消息类中信号消息提及符“@”数量的平均值 |
| 所有消息平均提及数量 | 候选消息类中所有消息提及符“@”数量的平均值 |
| 用户特征  （共18类） | 用户平均注册时间 | 候选消息类中所有消息作者用户的平均注册时间 |
| 发布平均流逝时间 | 候选消息类中所有消息的发布时间与作者注册时间差值的平均 |
| 用户平均点赞数量 | 候选消息类中所有消息作者的平均点赞（favorite）数量 |
| 用户平均关注者数量 | 候选消息类中所有消息作者的关注用户数量的平均值 |
| 用户平均被关注数量 | 候选消息类中所有消息作者被别人关注次数的平均值 |
| 用户平均影响力 | 消息类中所有消息作者被关注数量与关注者数量比值的平均值 |
| 用户平均消息数量 | 候选消息类中所有消息作者发过的消息总数的平均值 |
| 用户平均url数量 | 候选消息类中所有消息作者的个人资料是否附带url的平均值 |
| 用户平均简介数量 | 候选消息类中所有消息作者的资料中是否有个人简介的平均值 |
| 用户简介平均字符长度 | 候选消息类中所有消息作者的个人简介的字符长度平均值 |
| 用户简介平均单次长度 | 候选消息类中所有消息作者的个人简介的单词数量平均值 |
| 用户平均时区偏移 | 候选消息类中所有消息作者所在地区的utc时区偏移平均值 |
| 有影响力用户数目 | 候选消息类所有消息作者中影响力因子超过阈值的用户数目 |
| 普通用户数目 | 候选消息类所有消息作者中影响力因子低于阈值的用户数目 |
| 有影响力用户比例 | 比值：有影响力用户数目 / 候选消息类用户数目 |
| 用户平均转发数目 | 候选消息类所有消息的作者的转发消息总数的平均值 |
| 用户平均原创数目 | 候选消息类所有消息的作者的非转发消息总数的平均值 |
| 用户平均转发比 | 所有消息作者的转发消息总数与其发布消息总数比值的平均值 |
| 消息特征  （共8类） | 消息总数 | 候选消息类中所有消息的总数 |
| 消息平均问号数目 | 候选消息类中所有消息的问号数量的平均值 |
| 消息平均叹号数目 | 候选消息类中所有消息的叹号数量的平均值 |
| 消息平均情感得分 | 候选消息类中所有消息的情感评估得分的平均值 |
| 正情感消息比例 | 候选消息类中所有被认为是正面情感消息的消息所占比例 |
| 负情感消息比例 | 候选消息类中所有被认为是负面情感消息的消息所占比例 |
| 消息平均正向词数目 | 候选消息类中所有消息的情感正向词数目的平均值 |
| 消息平均负向词数目 | 候选消息类中所有消息的情感负向词数目的平均值 |
| 传播特征  （共4类） | 转发树根节点数目 | 候选消息类中所有的转发消息树的根节点总数 |
| 转发树非根节点数目 | 候选消息类中所有的转发消息树的非根节点总数 |
| 转发树最大深度 | 候选消息类中所有的转发消息树深度的最大值 |
| 转发树最大分叉数目 | 候选消息类中所有的转发消息树的最大分叉数目的最大值 |

其中原系统相关的特征有15类（去除了相关研究中不常用的单词熵特征，同时将长度特征具体化为字符与单词长度两种）。

插图索引

[图 1.1 推特中的误传消息 2](file:///E:\rumor\DataProcess\综合论文训练-钟仰新.docx#_Toc452860738)

[图 1.2 推特中的虚假消息 3](file:///E:\rumor\DataProcess\综合论文训练-钟仰新.docx#_Toc452860739)

[图 1.3 非谣言例子：自然语言分析面临的困难 4](file:///E:\rumor\DataProcess\综合论文训练-钟仰新.docx#_Toc452860740)

[图 1.4 谣言例子：谣言判定面临的困难 5](file:///E:\rumor\DataProcess\综合论文训练-钟仰新.docx#_Toc452860741)

[图 1.5 两则讨论同一个谣言的消息 6](file:///E:\rumor\DataProcess\综合论文训练-钟仰新.docx#_Toc452860742)

[图 2.1 原系统流程图 10](file:///E:\rumor\DataProcess\综合论文训练-钟仰新.docx#_Toc452860743)

[图 2.2 原系统使用的匹配模式 10](file:///E:\rumor\DataProcess\综合论文训练-钟仰新.docx#_Toc452860744)

表格索引

[表 3.1 数据集A的聚类结果（指标：NMI） 20](#_Toc452928637)

[表 3.2 数据集B的聚类结果（指标：NMI） 21](#_Toc452928638)

[表 3.3 各聚类方法在两数据集下最优的权重配比 24](#_Toc452928639)

[表 3.4 使用统计平均配比的加权相似度的聚类结果（指标：NMI） 25](#_Toc452928640)

[表 3.5 不同的预定义类数K的聚类效果（指标：NMI） 27](#_Toc452928641)

[表 3.6 消息举例：新老系统聚类对比 28](#_Toc452928642)

[表 4.1 新系统使用的特征列表 37](#_Toc452928643)



参考文献

[1] Cao, N., Shi, C., Lin, S., & Lu, J. (2015). TargetVue: Visual analysis of anomalous user behaviors in online communication systems. IEEE Transactions on Visualization and Computer Graphics, 22(1), 280-289.

[2] Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. Proceedings of the 20th International Conference on World Wide, 675-684.

[3] Karamchandani, N., & Franceschetti, M. (2013). Rumor source detection under probabilistic sampling. International Symposium on Information Theory, 2184-2188.

[4] Kwon, S., Cha, M., Jung, K., Chen, W., & Wang, Y. (2013). Prominent features of rumor propagation in online social media. IEEE 13th International Conference on Data Mining, 1103-1108.

[5] Qazvinian, V., Rosengren, E., Radev, D. R., & Mei, Q. (2011). Rumor has it: Identifying misinformation in microblogs. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 1589-1599.

[6] Seo, E., Mohapatra, P., & Abdelzaher, T. (2012). Identifying rumors and their sources in social networks. Proceedings of the Society of Photographic Instrumentation Engineers, , 8389

[7] Sun, S., Liu, H., He, J., & Du, X. (2013). Detecting event rumors on sina weibo automatically. Proceeding of the Web Technologies and Applications, 120-131.

[8] Takahashi, T., & Igata, N. (2012). Rumor detection on twitter. Joint 6th International Conference on Soft Computing and Intelligent Systems and 13th International Symposium on Advanced Intelligent Systems, 452-457.

[9] Wu, K., Yang, S., & Zhu, K. Q. (2015). False rumors detection on sina weibo by propagation structures. IEEE 31st International Conference on Data Engineering, 651-662.

[10] Yang, F., Liu, Y., Yu, X., & Yang, M. (2012). Automatic detection of rumor on sina weibo. Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, Article No. 13.

[11] Zhao, Z., Resnick, P., & Mei, Q. (2015). Enquiring minds: Early detection of rumors in social media from enquiry posts. Proceedings of the 24th International Conference on World Wide Web, 1395-1405.

[12] Hartigan, J. A., & Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. Applied statistics 100-108.

[13] Girolami, M. (2002). Kernel methods for pattern analysis. IEEE Trans. on Neural Networks, 13(3), 780-784.

[14] Johnson, S. C. (1967). Hierarchical clustering schemes. Psychometrika, 32(3), 241-254.

[15] Chen, W.-Y., Song, Y., Bai, H., Lin, C.-J., & Chang, E. Y. (2011). Parallel spectral clustering in distributed systems. PAMI 33(3), 568-586.

[16] Ratinov, L., & Roth, D. (2009). Design challenges and misconceptions in named entity recognition. Proceedings of the 13th CoNLL, 147-155.

[17] Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43rd ACL, 363-370.

[18] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 4(5), 513-523.

[19] Strehl, A., & Ghosh, J. (2003). Cluster ensembles - a knowledge reuse framework for combining multiple partitions. JMLR, 3, 583-617.

[20] Rus, V., Lintean, M., Banjade, R., Niraula, N., & Stefanescu, D. (2013). SEMILAR: The Semantic Similarity Toolkit. Proceedings of the 51st ACL, 163-168.

[21] Corley, C., & Mihalcea, R. (2005). Measuring the Semantic Similarity of Texts. In Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, 13-18.

[22] Rus, V., & Lintean, M. (2012). A Comparison of Greedy and Optimal Assessment of Natural Language Student Input Using Word-to-Word Similarity Metrics. Proceedings of the 7th Workshop on Building Educational Applications Using NLP, 157-162.

[23] Miller, G. A., Beckwith, R., Fellbaum, C. D., Gross, D., & Miller, K. (1990). WordNet: An online lexical database. Int. J. Lexicograph, 3(4), 235-244.

[24] Pocock, A. C. (2012). Feature selection via joint likelihood. Ph.D. dissertation, University of Manchester.

[25] Kohavi, R., & John, G. (1996). Wrappers for feature subset selection. Artificial Intelligence, special issue on relevance, 97(1-2), 273-324.

[26] Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and minredundancy. IEEE Trans. Pattern Anal. Mach. Intell., 27(8), 1226-1238.

[27] Kira, K., & Rendell, L.A. (1992). The feature selection problem: traditional methods and a new algorithm. Proceedings of AAAI-92, 129-134.

[28] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2(11), 559-572.

[29] Spearman C. (1904). The proof and measurement of association between two things. American Journal of Psychology, 15, 72-101.

[30] Paninski, L. (2003). Estimation of Entropy and Mutual Information. Neural Computation, 15(6), 1191-1253.

[31] Cardie, C. (1993). Using Decision Trees to Improve Case-Based Learning. Proceedings of the 10th ICML, 25-32.

致 谢

感谢张慧老师对我论文的知道。感谢黄经纬同学和我进行的有关光场的讨论，给我的工作拓展了思路。

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 外文资料的调研阅读报告

1 引言

近年来社交网络如Twitter、新浪微博广泛流行，这类平台的功能随着发展而逐渐扩大，从一开始的朋友间的日常分享、想法交流，到了今天广泛被个人、团体用于消息发布、宣传和推广，社交网络的“社会性”得到了前所未有的扩大。与此同时，社交网络也带来了很多负面的影响，谣言的传播就是其中之一。在此类平台上并不是所有消息都是真实的，部分用户会有意地或者无意地发布一些虚假信息、看似真实的误导信息，抑或是误报信息以及未经证实的可疑信息。这类信息的特点是真实性为假或者待定，但在其发布最初容易被误认为是真实信息，同时其真实性难以被迅速鉴定，我们称这样的信息为“谣言”。有些谣言的传播止于小范围的用户群体，而另一些谣言却由于其内容的轰动性、真实度的隐蔽性，以及平台用户的盲从性得到了广泛的传播。广泛传播的谣言经常会带来不同程度的负面影响：对个人或者团体的形象、利益损害，对大众价值观或认知的错误导向，甚至会引起大面积的恐慌或混乱。而社交网络每天发布的信息量巨大，因此监管部门不可能对消息进行逐条的人工审查，这使得谣言的检测与监控面临巨大的挑战。因此，利用机器来进行社交网络上谣言的自动检测成为了巨大需求。

在这篇调研阅读报告中，作者总结了近年来利用数据挖掘、机器学习等技术进行社交网络中谣言检测的相关研究，对他们进行分类说明。总的来说，这些研究利用了谣言的内容、传播特征等，或是训练分类器对社交网络上的消息进行“是不是谣言”的判断识别，或是提出一套评分机制来筛选出高可疑度的消息或话题名单，由于名单上候选的消息数量有限，这使得后续的人工检测成为了可能。

调研阅读报告的剩余部分将按照相关研究的分类展开，具体如下：章节2将介绍异常检测技术，章节3将介绍谣言源头检测技术，章节4将详细介绍谣言检测中的监督学习技术，章节5将重点说明谣言检测中的筛选排名系统，章节6将进行总结，章节7是本报告的参考文献。

2 异常检测技术

在社交网络中，谣言的散布一定程度上可以视为一种用户的异常行为，而一部分相关研究主要集中于社交网络上的异常检测。Nan Cao等人在他们的文章中提到这类研究一般包括两种：使用带标签（正常vs异常）的训练数据进行有监督的学习，或者提出不需要训练数据的非监督学习模型[1]。在他们的文章当中，他们使用的是一种无监督的学习技术，TLOF模型，这个模型有以下优点：1.这是无监督的学习，因此不需要训练数据；2.此模型考虑了用户的历史行为，而不仅仅是社交网络中某个时刻的情况，因此能更准确地检测出“异常”；3.此模型并非只是给用户进行分类（打上“正常”或“异常”的标签），其还能给出一个“异常”程度的分数评估，这就方便给用户进行排序，筛选出行为最异常的用户名单；4.此模型基于欧几里得进行异常测量，方便于对结果进行可视化（事实上他们设计了一套先进的可视化系统方便监测社交网络上的异常用户）；5.此模型有很高的效率，其时间复杂度仅有O(NlogN)，N为用户数量。

在TLOF模型中，异常检测的核心是一种名为LOF的度量机制，此机制基于于欧几里得空间，假设每个用户都被表示为欧几里得空间中的坐标点，此度量对某个坐标点综合考虑了它距离其K近邻邻居点的远近，以及K近邻邻居点距离它们各自的邻居的远近，计算出所谓的“近邻密度”，从而估算出该坐标点是否属于以及有多大程度属于“离群点”[1]。Nan Cao等人在模型中综合考虑了该用户当前时刻的LOF度量在所有时间的所有用户的数据中的离群程度，以及该用户当前时刻的LOF度量与他自己过去一段时间的历史记录相比的离群程度，进而估算出其当前时刻异常程度的总评分数。他们的研究是通过对每个用户抽取不同的特征向量，形成一个高维的向量空间，在此空间中使用LOF度量。其抽取的特征包括用户的行为特征（发布率、回复率）、内容特征（话题类型、情感评分等）、互动特征（传播消息的广度）、时间特征（发布或回复的时间间隔以及熵等）、网络特征（好友网络出度入度的突变）、用户简历特征（修改用户名的频率等）。

如前所述，这类研究检测的重点是用户的异常行为。而谣言识别经常是需要检测出被广泛传播的谣言或虚假消息，虽然也需要检测出是哪些用户发布谣言以及在谣言的传播中起关键作用，但其研究的直接对象是社交网络的消息或话题，而并非用户。第二，“异常”并不等同于“谣言”，在异常检测中会利用“离群度”来衡量可疑程度，但请试想如果一个谣言被认为是真实消息，从而被广泛传播，那么在短期一段时间中其特征与真实新闻将类似，从而会有较低的离群度而导致无法被识别出来。最后，在此类研究中会将用户的当前时刻行为在历史记录中的离群度作为异常的度量因素之一，但设想一个经常散布谣言的用户，当前行为相较于历史记录的离群度并不能作为衡量“异常”或“谣言”的标准。因此，异常检测技术与谣言识别技术虽然相关，但也有明显差异。

3 谣言源头检测技术

在相关研究中，有一类工作着重于谣言的源头检测，这类工作一般会在社交网络上设置一些“监测节点”（实际上也就是某些用户），让这些节点汇报他们是否在社交网络上听到过某条特定的消息或谣言，然后根据监测节点的汇报情况以及社交网络中的follower-followee关系，用有向图对信息流动建模，进而分析出消息或谣言的源头，或者传播中心。

Ensoo Seo等人在他们的研究中就是采取了这样的方式[6]。他们将监测节点分成两类：正节点（听过谣言的节点）和负节点（没有听过谣言的节点），然后分析每个节点到这两类节点的可达性和总距离，对他们进行排序，找出最可能为谣言源头的节点（用户）。他们基于的基本假设是，如果网络中仅有一个谣言源头，那么它将到所有正节点都是可达的，如果存在多个这样的节点，那么其中到所有正节点总距离越小的节点越可能是源头；理想状态下源头节点到所又负节点都是不可达的，也就是及源头节点到所有负节点的距离都是无穷，而实际分析中的原则是到所又负节点的距离越远越可能是源头节点。而在具体实现上，他们综合考虑了这四个因素对节点进行评分和排序。

Ensoo Seo等人在该工作中同时提出了判断某消息是不是谣言的一种算法。他基于的基本假设是，谣言应该是单一源头或者源头节点数量很小，而真实的消息应该是多源头一起发布、相互佐证。为了选出所有可能的源头节点集，他们在文章中提出了一种贪心算法：每次从网络中找出一个节点，此节点在正节点集中拥有最多的可达节点，把此节点加入源头候选集中，再将它在正节点集中的可达节点都从正节点集中去除，然后进行下一轮筛选，直到正节点集为空。此源头候选集中节点数目越小，则该消息越可能是谣言。此外，为了增加算法准确性，他们提出应当将候选集每个节点到正节点集每个可达节点的总距离作为第二指标，距离越小，则该消息的真实源头数目可能越少，即越可能是谣言。

Nikhil Karamchandani在他们的研究中允许监测节点汇报他们听过某消息的可能性（介于0和1之间的概率，称“感染概率”），而不是他们是否明确听过某消息[3]。另外，他们建立的是无向图模型，更准确地说是在正则树和非正则树两种模型下，通过图中每个时刻节点的感染概率以及节点的连结关系，计算出谣言的“边界”，从而估算出每个节点的“中心度”，从而计算出最有可能的谣言源头。

此类研究的重点在于通过“监测节点”的感染状态和社交网络的图结构，试图还原出谣言的传播轨迹，从而找到最有可能是谣言散布源头的用户，而谣言识别则是着重于分析出哪些消息是谣言，而哪些不是，因此两者在目标和方法上都有相当的区别。虽然Ensoo Seo等人也提出了一种谣言检测的算法，但此算法仅考虑了消息的网络传播特征，而实际上还有很多有价值的特征如内容特征、用户特征等也可以作为谣言识别的考虑因素，此外也还有很多其它有价值的网络特征也值得参考，因此这种算法应当还有很大的提升空间。但是此类研究对于谣言识别技术有相当的参考价值，包括设置监测节点的策略，以及图分析和网络特征的考量都非常有借鉴意义。

4 谣言检测中的监督学习技术

在谣言检测中，有很大比例的研究都采用了监督学习技术。这类研究致力于选择或提出更有效的特征，包括消息的内容特征、传播特征、消息发布者的特征等等，然后使用有标记的社交网络消息数据集，尝试使用、训练出各种分类器，使用交叉验证进行评估，找出分类准确度更优的方案。

Vahed Qazvinian等人的研究不仅致力于检测出谣言，还致力于检测出某消息是在传播或同意谣言的内容，还是质疑、反对它，他们将后一步的目标称为“置信分类”[5]。其选择的特征包括：内容特征，包括消息中的单词（unigram）和二元词组（bigram）在正负样本中的出现情况；网络特征，具体来说是消息发布用户以及消息转发源头用户发布正负样本的比例，实际上也就是消息发布者和被转发者的“信用评价”；Twitter特有特征，包括url和hashtag两类，对附有url的消息抓取url网页内容，然后对其中的单词和二元词组出现情况抽取特征，对附有hashtag的消息，抽取hashtag的单词出现情况作为特征。而在分类器方案上，他们采取的是对每类特征建立贝叶斯分类器，最后将每个分类器的得分线性组合起来作为最终的分类得分，在监督学习时利用训练数据学习出最优的线性函数（即各分类器的权重）。

此工作作为最早的社交网络谣言检测研究之一，其选择的特征比较简单朴素，数量也较少，但是对特征的选取和基本分类却给了后来的相关工作很好的启示和范例作用；其选择的监督学习技术属于朴素贝叶斯和线性回归，也相对简单，有提升空间；其对Twitter数据的收集和标记方式对后来的研究也有借鉴作用，但是比较遗憾的该研究只选取了5个谣言话题，而且在实验设计上存在一定缺陷，每次训练的分类器只是针对某一话题的谣言进行训练和评估，也即这是个狭义的分类器，即使其实验结果达到了90%的F1-score，但该实验并没有表明其具备检测广泛话题的谣言的能力。

Fan Yang等人的研究借鉴了Vahed Qazvinian等人的研究，但研究的对象从Twitter转移到了新浪微博[10]。他们借鉴并拓展了Vahed Qazvinian等人研究的特征选取，修改补充了特征种类，并加入了新浪微博特有的特征：内容特征上考虑了正负情感词语的数目，以及消息是否附有URL和多媒体链接等；传播特征上考虑该消息是原创的还是转发的，以及其评论和转发数目；用户特征上考虑了用户是否实名认证、性别、有无个人描述、头像类型、用户名类型、注册时间、粉丝数量、关注数量、所发微博数量等。另外，他们提出了两种“有效的”新特征，包括目标微博的发布所在地以及发布所用的客户端类型（是移动设备APP，还是网页客户端等），他们的研究发现谣言中有很大比例是使用非移动设备客户端发布的，而且是在国外发布的。而在分类器的选择上，他们使用了SVM技术，核函数采用了径向基函数（RBF）。

此研究的数据收集部分很有参考价值，由于研究的对象是新浪微博，研究者有效地利用了一个新浪微博官方账号“微博辟谣”作为谣言的ground truth收集标准，此官方账号的功能就是对一些传播广泛的微博谣言进行辟谣，因此研究者收集了此账号从2010年到2012两年间的辟谣话题，并使用微博API收集话题相关微博，最后再通过两个标记者人工标记相关微博是支持谣言还是质疑或反对谣言。这样的数据收集和标记方案大幅度的增加了数据集中谣言话题的数目和类型广度，而且在实验设计上该研究并非针对单一话题进行分类，而是对所有话题的数据集进行是否谣言（或说支持谣言）的分类，虽然最后的F1-score仅有78%，但实验表明该分类器可以被用于广泛话题的谣言检测。

Shengyun Sun等人的研究对象也是新浪微博中的谣言，他们借鉴了Fan Yang等人的数据收集方式，也是通过“微博辟谣”账号来收集谣言话题（时间范围是2011下半年，共104个谣言微博），但他们的研究着重于事件谣言（谣言内容是某件事情）的检测[7]。在Fan Yang等人研究的特征选取基础上，他们又进行了扩充：内容特征加入考虑了“@”符号、雷同微博数目、微博中关于事件的动词的数量，以及是否含有强否定词；用户特征加入考虑了“声望”的概念（用户粉丝数目占用户关注加粉丝数目的比例），用户发布的所有微博中含有事件动词的微博比例以及含有强烈否定词的微博比例；他们原创性地加入考量了多媒体（图片）特征，对于带图片的微博，他们利用搜索引擎搜索该图片，没有结果（原创图片）视为一类，如果有结果，则计算图片时间戳与微博时间戳的时间跨度并以某阈值进行二分类，以此作为三类特征；此外他们进一步利用搜索引擎，提出了一种检测图文无关谣言的方法。

在实验中他们尝试使用朴素贝叶斯、贝叶斯网络、神经网络和决策树作为分类器进行训练，实验表明他们提出的时间动词特征、强否定词特征、以及图片时间跨度特征对分类准确度的提升有明显帮助，其中贝叶斯网络和神经网络的F1-score最高（都接近74%）。由于使用了“微博辟谣”进行数据收集，所以他们数据集中的谣言话题也具有多样性，训练出来的分类器也能检测出广泛话题中的谣言，唯一的不足是他们的数据仅仅收集自104个谣言话题的参与用户在2011下半年的所有微博，而且在数据集中正例（谣言微博）的比例只有不到0.4%。

Carlos Castillo等人在他们的工作中对Twitter数据设计了两个分类任务，一是消息是否具有新闻价值，二是消息是否可信[2]。他们的第二个任务实际上也就是谣言检测[2]。此工作的难得之处在于虽然它是最早的谣言检测研究之一，但它考虑的特征种类非常多，共68种基本分为四类：消息特征、用户特征、话题特征、传播特征。其中消息特征考虑了很多符号（问号、叹号、表情符）、消息长度以及情感得分等；话题特征则是同类推特中各类消息特征和用户特征的比例；传播特征考虑的是消息传播树的出度、大小、深度等因素。他们在文章中尝试过使用SVM、决策树、决策规则和贝叶斯网络作为分类器，称这几类分类器有近似的分类准确度，其中最好的是J48决策树（89%）。而位于J48决策树顶层的特征有：话题特征中，含URL的推特比例、含强否定词的推特比例以及含感叹号的推特比例；用户特征基本都位于顶层，发布过的推特数目和好友数目最为重要；传播特征中，传播树中最大层的节点数目位于决策树顶层。

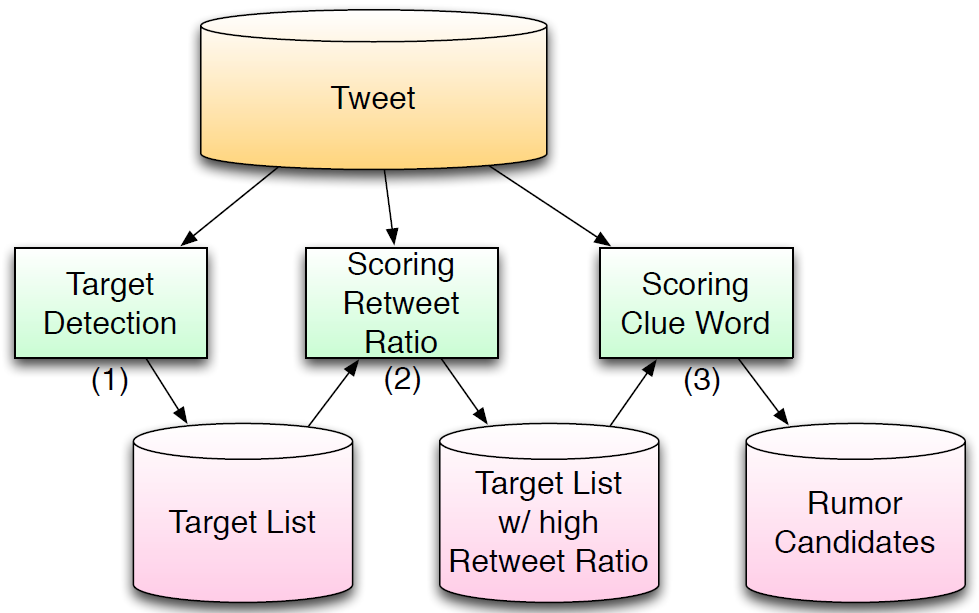
Sejeong Kwon等人的研究提出了一种基于时间序列分析的特征选取，他们基于的基本假设是谣言和非谣言相关的推特数目表现在时间轴上会有诸如峰值强度、峰数目、峰周期、反应延迟等的差异[4]。因此，他们提出了一种名为PES（Periodic External Shocks）的模型去拟合某话题推特数目与时间的函数关系，并采用参数学习技术学出该模型的10个相关参数，作为特征输入分类器，最终对该话题是不是谣言进行分类。在分类时它们还考虑了话题传播树的结构特征（传播特征）和语言特征（内容特征），尝试使用了SVM、决策树和随机森林作为分类器，最终随机森林的F1-score最高（接近90%），并分析出最有效的11个特征，其中包含了PES模型的3个参数特征，但语言特征还是占最高比例。值得赞许的是，他们将所使用的带标记Twitter数据集发布了出来。

Ke Wu等人在他们的研究对象是新浪微博中的谣言，他们提出了一种graph kernel的新度量作为特征[9]。具体做法是，对每一条微博提取出它的传播树（转发评论树），树上节点是转发的用户，将节点分成两类——普通用户或大V用户，分类标准是给用户的“声望”（粉丝数目对关注数目的比值）设置阈值，树上的每条边用一个长度为3的向量表示，分别是转发微博对原微博评论文字的赞许程度、质疑程度以及总的情感得分。他们的研究称过去的研究所选择的传播特征都是诸如出入度、深度、广度这样的广泛值而忽略了一些更具体的传播特征，他们相信消息在普通用户和大V用户间的流动方式对谣言的检测将有很大帮助。但由于传播树巨大，Ke Wu等人提出了一套规则来合并传播树中满足条件的同类节点，以及他们的边，得到一棵约减的传播树，最后他们提出一种基于随机游走的度量计算两棵传播树的相似度，并使用相关急速将计算复杂度降至O(n4i)，其中n是树的节点数目，i是迭代次数。他们在特征选取上除了上述的graph kernel还选择了如内容特征、用户特征等的传统特征，最终将这两者用权重进行配比作为混合内核，采用SVM技术训练分类器。而在“传统特征”中他们其实也提出了一些有参考意义的新特征，如使用LDA分析消息所属的话题分布形成话题特征，以及使用搜索引擎进行简单的谣言分析的搜索引擎特征等。他们的实验最终达到了90%的F1-score，实验显示graph kernel在所有特征中对分类准确率的提升帮助最大。但此graph kernel的问题在于其虽然经过加速，但复杂度依然很高，而且他们的方法需要对两两微博都进行相似度计算，也就是O(N2)的外循环复杂度，则训练集的数目不能过大，这与微博数目巨大、种类繁多的特点是矛盾的。

总的来说，此类工作不断研究社交网络中谣言的内容、发布、传播等各种特性，提出新的更有效的特征，并尝试使用各种分类器找到最优的分类方案。关于其局限性，谣言检测中的监督学习技术非常依赖于数据集的收集和标记质量，而社交网络中大规模带标记的高质量数据集却往往是难以获得的；虽然此类工作公布的分类准确度都普遍很高，甚至能达到90%，但其解决方案往往只能对消息进行逐条分类，即便是10%的错误率在大规模微博下也是非常难以接受的比例；再者，这类工作无法系统地检测出共同讨论一个话题的消息集，并利用消息集的特征判断该话题是否谣言；最后，此类研究的很多特征提取和训练过程都相当耗时，这在数据膨胀的社交网络中无疑面临着规模性的挑战。

5 谣言检测中的筛选排名系统

谣言检测还有一小部分研究提出的解决方案是筛选排名系统。这类研究多是对社交网络的消息数据集进行话题提取，然后对某一类话题的消息抽取特征并进行可疑度与重要度评分和排名，最后筛选出最有可能为谣言的话题候选名单，以此缩小谣言检测范围，最后将有限的候选名单；交予人工判断。

Tetsuro Takahashi等人在他们的研究中提出了一套筛选谣言话题的机制[8]。图1是他们所设计的系统，总共包含三个步骤：第一步，他们采用了一种名词实体提取技术，从所有Tweets的文本中提取出名词实体，然后对每个名词实体，计算出每天含该词的微博数的最大值，并设置最低阈值，筛选出高于阈值的那些实体词形成Target List，这样每个名词实体代表一个话题，这一步实际上做的是一种“爆发检测”，筛选出的是那些被大众热烈讨论的话题；第二步，他们对Target List中的话题做进一步筛选，选取出含有该名词实体的所有Tweets，并计算出其中属于Retweet的比例，并设置最低阈值进行过滤，这一步基于的基本假设是影响重大的谣言或话题都拥有很高的转发率；而第三步则是进一步筛选，计算出含有某名词实体的Tweets集中，含有“线索词”的Tweets比例，而经他们的研究分析，为了不失一般性，最后仅保守地选取了“false rumor”作为线索词，也就是计算出Tweets集中含“false rumor”这个词组的Tweets比例，然后设置最低阈值，筛选出高于阈值的话题作为最后的谣言候选名单，交予人工进行验证。最终他们在数据集中筛选出了10个候选话题，经人工检测其中真正属于谣言的有6个。

图A-1 Tetsuro Takahashi等人的系统

此研究虽然提出了一套筛选系统，但其采用的技术过于简单，设计上也存在很多问题：将实体名词作为话题的代表不尽合理，因为通常一个话题可能包括多个实体名词；仅用关键词的出现次数和retweet率作为筛选条件过于粗糙，并且只能筛选出一些被广泛讨论的问题；而关键的谣言检测步骤也仅是使用了一个线索词“false rumor”进行过滤，显然也是过于简单粗暴的，其提升空间很大；而他们的实验只汇报出前10名的候选话题的准确度，这样的评估也是远远不够的。但是他们的思路值得借鉴，整个系统首先经过“爆发检测”筛选出重要的讨论话题，然后利用谣言的特征线索作进一步筛选或排名，这样的方案不需要训练数据集，执行效率极高，却能从Tweets集中检测出话题，并通过排序选取出最可能是谣言的候选名单。如果将其采用的话题检测、筛选和排名技术进行提升，很可能做出一套实用性、规模性很高的系统。

Zhe Zhao等人的研究设计出的筛选排名系统具有高规模性，并可以用于谣言的早期检测[11]。图2是他们的系统，一共分为五步：第一步，他们先从Tweets集中检测出反映谣言的“信号tweets”，他们的基本假设是在谣言的传播过程中或多或少会伴有质疑、反对或纠正的tweets，这类tweets就是谣言话题的“信号tweets”，先检测出此类tweets可以很大程度帮助谣言检测，而他们采用的检测的方法是模式匹配，图3是他们用于识别信号tweets的模式规则，前三个是质疑的模式，后两个是反对和纠正的模式，如果某个tweets满足其中一个模式则将它识别为信号tweet；第二步，在识别出所有信号tweets之后，对他们进行聚类，该研究非常重视tweets的“重叠率”，他们认为识别讨论同一话题tweets的步骤应当做得尽量谨慎，因为谣言的传播有很大程度都是通过retweet进行的，在此过程中原始tweet和retweet的内容重叠率应当很高，并且由于微博数量众多，聚类的步骤的复杂度应当尽量低，因此他们对信号tweets两两计算内容的单词、二元词组以及三元词组的集合重叠率（Jaccard coefficient）并设置一个较高的最低阈值（0.8），高于此阈值则将两信号tweets聚为一类，此聚类过程有点像图模型中的community detection，由于阈值设置较高，所以每个cluster不会过大，然后筛选出其中规模超过一定阈值cluster；第三步，对每个信号cluster提取出现率较高的单词、二元词组和三元词组作为其statement集合，此集合表示的就是每个cluster的真正的话题内容；第四步，将非信号tweets逐一与每个信号cluster的statement集合进行重复率计算，将超过一定阈值的非信号tweets聚类到该信号cluster中，形成一个话题cluster；第五步，对话题cluster中的信号和非信号tweets提取特征，并采用有监督学习技术训练出一个可疑度排名器，利用此排名器对话题cluster进行可疑度排序，找出最可疑的话题作为谣言候选列表交予人工检测。

图A-2 Zhe Zhao等人的系统

图A-3 信号tweets检测采用的模式规则

值得注意的是此研究中的第五步用到了有监督学习技术，这表明此系统仍需要一定的标记训练集，但为了规模性，该研究选取的特征都比较简单，如信号tweets占所有tweets的比例、话题cluster中词分布的混乱程度（熵）、tweets的平均长度、retweet所占比例以及所含url、hashtag和mention的数目等，并没有第四章的监督技术选取的特征那么复杂和有效。其尝试采用的分类器（排名器）是SVM和决策树，最终发现决策树的准确度更高。最后实验表明，该系统能从每天产生的tweets中筛选出三百个左右的谣言候选话题，大大减少了人工筛选的工作量，其排名前10的候选者中真正的谣言有60%，而前50的候选者中真正的谣言有33%，而所有候选者（共350个）的准确率有26%。相较于采用监督学习的谣言检测技术（最高90%的F1-score），这个准确率显得十分低，但是监督学习技术采用的分类器仅能对每条tweets进行分类，所以错误率即便仅有10%，在巨大的tweets基数下也是不能容忍的；相对的，此系统能对tweets进行话题检测，将一个月的12亿条tweets缩减为350个候选话题，并最终有92个是真正的谣言，其实用性远远高于前者；由于该系统的核心检测步骤是对信号tweets的检测，又因为此研究发现谣言消息的信号tweets都出现得特别早（一般10min到4h就出现），所以此系统可用于谣言的早期检测（此结论经过了实验说明）。最后，由于系统采用的算法复杂度不高，所以其可用于真实的社交网络大规模消息的谣言检测（该系统处理每天产生tweets的10%只需要30min）。

当然，目前该系统使用的聚类算法、信号检测技术、话题抽取技术和可疑度排名技术都较为简单，如果舍弃一定的规模性，采用更精细的技术，比如借鉴第四章监督学习技术中抽取的特征和分类器选取，设计出更好的排名器，那么很有可能提升其候选集的准确度，得到更为让人满意的结果。

总结而言，谣言检测中的筛选系统一般采取的技术都很简易，系统设计也依赖于很多谣言相关的先验知识和观察（如检测信号tweets的模式规则），准确率也不及监督学习技术，但其具有高效率、高实用性的特点。

6 总结

本报告一共介绍了与谣言检测相关的四类技术，其中相关度较高的是监督学习技术以及筛选排名系统。监督学习技术有高准确性，但依赖于高质量的带标记训练集，其算法复杂度通常很高，而且分类只能逐条消息进行，实用性低；而筛选排名系统虽然准确度低而且最后需要一定的人工检测，但它能检测出可能的谣言话题，并对可疑度进行排名，其规模性与实用性都很高。如果能参照筛选排名系统的设计思路，同时借鉴监督学习的特征提取和分类技术，将两者的长处结合起来，那么很可能会取得更好的成果。

最后，社交网络的谣言检测目前还处于不成熟阶段，相关研究数目极少，也还有很多没有尝试引入但可能有效的技术（如半监督、无监督学习，模式识别等），因此有待相关学者进行更多有价值的研究，使该领域的理论更加丰富。

7 参考文献

[1] Cao, N., Shi, C., Lin, S., & Lu, J. (2015). TargetVue: Visual analysis of anomalous user behaviors in online communication systems.*IEEE Transactions on Visualization and Computer Graphics, 22*(1), 280-289.

[2] Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. *Proceedings of the 20th International Conference on World Wide,*675-684.

[3] Karamchandani, N., & Franceschetti, M. (2013). Rumor source detection under probabilistic sampling. *International Symposium on Information Theory,*2184-2188.

[4] Kwon, S., Cha, M., Jung, K., Chen, W., & Wang, Y. (2013). Prominent features of rumor propagation in online social media. *IEEE 13th International Conference on Data Mining,*1103-1108.

[5] Qazvinian, V., Rosengren, E., Radev, D. R., & Mei, Q. (2011). Rumor has it: Identifying misinformation in microblogs. *Proceedings of the Conference on Empirical Methods in Natural Language Processing,*1589-1599.

[6] Seo, E., Mohapatra, P., & Abdelzaher, T. (2012). Identifying rumors and their sources in social networks. *Proceedings of the Society of Photographic Instrumentation Engineers, , 8389*

[7] Sun, S., Liu, H., He, J., & Du, X. (2013). Detecting event rumors on sina weibo automatically. *Proceeding of the Web Technologies and Applications,*120-131.

[8] Takahashi, T., & Igata, N. (2012). Rumor detection on twitter. *Joint 6th International Conference on Soft Computing and Intelligent Systems and 13th International Symposium on Advanced Intelligent Systems,*452-457.

[9] Wu, K., Yang, S., & Zhu, K. Q. (2015). False rumors detection on sina weibo by propagation structures. *IEEE 31st International Conference on Data Engineering,*651-662.

[10] Yang, F., Liu, Y., Yu, X., & Yang, M. (2012). Automatic detection of rumor on sina weibo. *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics,*Article No. 13.

[11] Zhao, Z., Resnick, P., & Mei, Q. (2015). Enquiring minds: Early detection of rumors in social media from enquiry posts. *Proceedings of the 24th International Conference on World Wide Web,*1395-1405.

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** | **考核组组长签字：**  **年 月 日** | | | | |
| **指导教师评语** | **指导教师签字：**  **年 月 日** | | | | |
| **评阅教师评语** | **评阅教师签字：**  **年 月 日** | | | | |
| **答辩小组评语** | **答辩小组组长签字：**  **年 月 日** | | | | |

**总成绩：**

**教学负责人签字：**

**年 月 日**

1. http://www.mathworks.com/matlabcentral/fileexchange/26182-kernel-kmeans，2016年3月13日的版本 [↑](#footnote-ref-1)
2. http://cn.mathworks.com/help/stats/hierarchical-clustering.html，Matlab R2014b版本 [↑](#footnote-ref-2)