清 华 大 学

综 合 论 文 训 练

题目：社交网络中的谣言检测

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 6 月 12 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：钟仰新　导师签名：刘世雷　日　期：2016.6.12

中文摘要

社交网络（如推特、微博）中存在大量虚假的谣言，其传播可能对个人和社会造成巨大的危害。社交网络巨大的消息量使人工排查谣言的成本很高，因此谣言的自动检测技术具有重大理论研究和实际应用意义。

已有最新谣言检测方法主要存在两个方面的问题：候选话题重复率过高和谣言检测准确率过低。为了解决这些问题，本文围绕社交网络中的谣言检测问题，展开了谣言检测技术的探索和实现工作。在原算法框架的基础上，本文设计实现了一个更精确、实用的新谣言检测框架。本文首先探讨了 6 种适合社交网络话题聚类的相似度度量，并用加权平均的方式将它们结合在一起；其次，设计了一种“以过滤器指导起点的浮动式包装器”的特征选择方法选择出有效的谣言检测特征；最后，设计了一种基于多分类器投票思想的候选可疑度排名方案。

在真实数据集上的实验结果表明，新框架候选的话题重复率比原框架降低了 50%，检测出排名前 100 候选的谣言准确率提高了 63%，其中前 20 候选的准确率提高了 100%。

关键词：社交网络；谣言检测；聚类；特征选择；分类器

ABSTRACT

There are numerous rumors in social network (e.g., Twitter and Weibo). The rumor spread may have harmful effect on individuals and communities. The huge volumes of posts on social networks make the cost of manual checking extremely high. As a result, automatic rumor detection techniques are of great significance to theoretical studies and practical applications.

The state-of-the-art rumor detection method has two major problems: many detected candidate topics share the same meaning, and the precision of detection is not satisfying. To solve these problems, in this paper, we have developed a more precise and practical rumor detection method based on the state-of-the-art one. We first discussed 6 similarity metrics that are suitable for topic clustering in social network and combined them by weighted average. We have also designed a feature selection method called “float wrapper initialized by filters” to select effective features for rumor detection. Finally, to detect candidate rumors we have designed a rumor ranking method based on majority voting of multiple classifiers.

Experimental results on real-world datasets demonstrate that the new method can lower the rate of candidate overlap by 50%, improve the Top-100 precision of rumor detection by 63%, and improve the Top-20 precision by 100% particularly.

Keywords: Social network; rumor detection; clustering; feature selection; classifier

目录

第1章 引言

1.1 社交网络中的谣言

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，顾名思义将人们日常的社交活动，推广拓展到了互联网中；它是一类由互联网公司提供的能让用户在线交友、交流、分享的网络平台。

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”，人们通常称之为“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。此消息本身并无恶意，是想鼓励人们勤洗手来预防埃博拉病毒，事实上用户很可能是看到了世界卫生组织提倡洗手预防埃博拉的新闻，然后发布了这则消息；但是用户却误解了世界卫生组织的意思——肥皂洗手的确能一定程度预防埃博拉，但不是因为这样能杀死病毒，而是因为肥皂加流水能有效减少手上附着的病毒数目（参考信

图 1.1 推特中的误传消息

（图片来源：https://twitter.com/funny_truth/status/541155156264382465）

息来源：http://www.killebolavirus.com/does-hand-sanitizer-kill-the-ebola-virus/），而世界卫生组织也从未在任何场合提到一般的肥皂能有效杀死埃博拉病毒。

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，它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性质。由于社交网络的用户量庞大、用户网络连通性强，一旦传播起来，虚假消息在短短几小时内就能被千万人知晓。这类消息的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 这则消息谈论的就是一则虚假消息：美国摇滚歌手 Akon 在刚果的一场表演中将自己包裹在一个巨型的泡泡球内，谣言称 Akon 此举的目的是为了防止在与群众发生接触，从而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，对 Akon 的名声造成了很大的冲击。但后续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使演出更加有趣，增进与观众的互动；事实上他过去几年已多次在世界各地进行这种形式的表演（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），原谣言缺乏论据。

由于社交网络每天传播的大量消息中混杂了不少诸如此类的谣言消息，如果不加以监督管制，及时发现和组织高危害谣言的传播，那么社交网络将成为虚假消息滋生的温床，将对社会产生不可估量的负面影响。

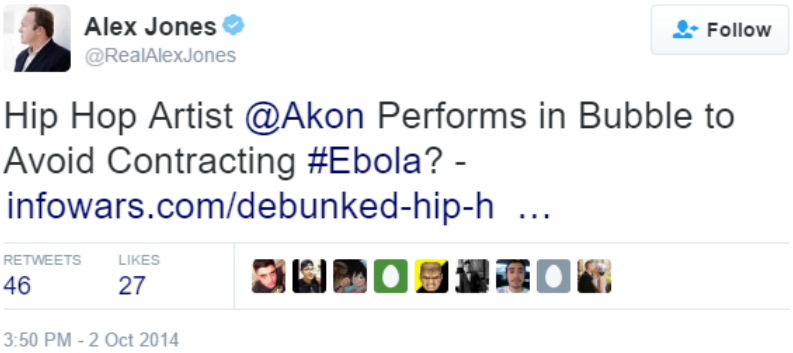

图 1.2 推特中的虚假消息

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

1.2 谣言检测技术与其面临的挑战

因为社交网络存在不少谣言，所以需要找到一种有效的监管机制。一种朴素的想法是设立“消息审查员”的职位，利用人力去审核消息，找到其中的谣言。但是社交网络的消息数量巨大，人工地进行逐条审查并不实际。

正是由于人工检测成本太大，所以人们开始研究发展能自动识别、检测出谣言的技术，现统称为“谣言检测”技术。这类技术通常是利用计算机强大的处理能力，提取分析大量消息的特征，评估其可疑度，进而自动找出疑似谣言的消息。

谣言检测技术在研发中面临了许多挑战，以下是其中两个主要的问题：

识别准确率不高，是谣言检测技术面临的一大问题。谣言检测中存在大量将非谣言的消息误认为谣言，以及将谣言误认为普通消息的现象，检测准确率不高。由于此类技术通常主要是通过分析消息的文本进行可疑度评估，而自然语言非常难以分析，这常成为谣言检测的瓶颈。如图 1.3 就是一个例子，在这则消息中，存在着如 panic（恐慌）、rumors（谣言）和 circulating（流传）这样的词语，如果仅仅分析它们各自的语义和情感导向，机器很容易误认为这是一则谣言相关消息。但事实上在 panic 这个词前面还存在词组 no reason（没有理由），一下子就将 panic 的词义进行了反转；而在 rumors 与 circulating 这些词前面还存在 get out（走出）这个词组，只有将这些词都串联起来，这个句子的真正意义才能浮现：它不是在讲一则引起恐慌的谣言，而是在讲我们不应恐慌，而要走出来面对那些传播的谣言。因此这仅是一条非谣言的普通消息。

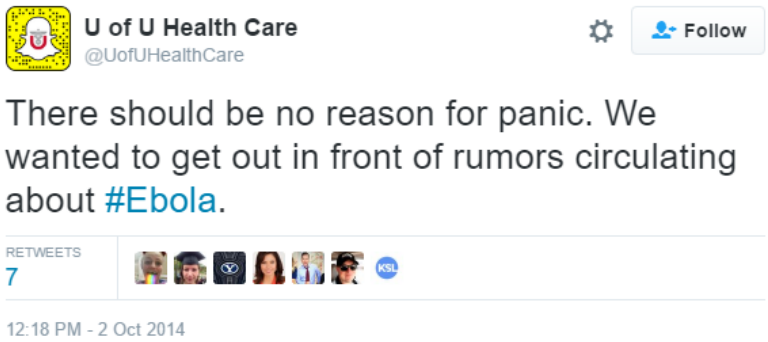

图 1.3 推特中一则非谣言的消息

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

此外，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——谣言的判定需要对消息的来源、内容的论据进行深入分析，这其实涉及到很多专业知识和技巧，不是单一的文本分析就能完成的事情。如图 1.4 中的消息就是一个例子。消息称感染了埃博拉病毒的护士 Nina Pham 的男朋友也出现埃博拉病患的症状，在文本后半部分作者还附上了用以佐证的新闻网址，点开后也的确是一则符合消息内容的网页新闻。现在读者能否告诉我这是一则正常的新闻推广还是一则谣言？大多数读者可能会认为是前者，因为消息带有一个正规新闻网站的链接以示消息来源。但事实上我们不能这么简单地下结论，因为即使是正规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。非常不巧，这则新闻正是这样的不实报道，后来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-

图 1.4 推特中一则谣言消息

（图片来源：https://twitter.com/Jody_Arrington/status/523633560779907074）

boyfriend-rumored-admitted-hospital-ebola-symptoms-alcon-releases-1707586）。这个例子说明，仅靠文本分析不能完成谣言检测的任务。

因此为了提高谣言检测的准确度，许多研究者不仅考虑文本语义方面的因素，还会综合考虑各方面的因素：如消息发布者影响力、消息来源可靠度、消息传播路径拓扑结构等等的一些特征，在这些特征上施以复杂的分析模型（如规则模型、

概率模型），最终形成较为健壮、识别准确率较高的谣言检测技术。但这就衍生出了更多相关的问题：如何找出那些有价值的特征？怎样的分析模型能提高框架的检测准确率？本文在第 4 章和第 5 章中将讨论这两个问题。

检测结果重复率过高，是谣言识别技术面临的另一大问题。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，假设几百个用户在同时讨论、转发一则谣言，即使这些消息都能被检测识别出来，但当我们将识别结果交给审查人员进行人工复核时，问题就出现了：这么多的谣言相关消息，审查人员根本看不过来；更糟糕的是，有大量的消息都是关于同一个谣言的，本来这样的消息审查一两条就能确定谣言话题是什么，足够进行辟谣了，但候选集中却有几百条类似的消息，重复去审查这些消息不仅浪费审查员大量的时间，而且毫无意义。因此，要想利用有限的人力审查更多不同的谣言话题，就必须将那些讨论同一话题的消息归为一类，降低检测结果的重复率。

4chan is trying to spread some shit rumor about ebola. If you see the hashtag "#ebolaindoritos", it's horseshit. http://t.co/PSriKtBaoX
Also, a fine round of applause for rumormongering goes to these 4channers. #ebolaindoritos http://t.co/r0u6eCo3K5

图 1.5 两则讨论同一个谣言的消息

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特色的文本不适用或不够准确，因此需要开发出针对社交网络消息特点的聚类方法。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，但其文本相似度却很低：由于文本短小，导致了重叠的关键词不多，本来 4chan、rumor、ebolaindoritos 是三个共同的关键词，但是在第二则消息中 4chan 却变体成 4channers，而 rumor 则与 mongering 连在一起导致成为了另一个词，因此重复的关键词仅剩下了 ebolaindoritos 一个；另外文本附上的网页地址也完全不同（尽管它们是刊登在不同网站上的同一则新闻）。因此，如果从传统的聚类方法出发，这两个文本很可能不被归为一类。因此为了更准确地聚类，研究者们通常会加入一些社交网络特有的要素，去衡量两则消息的相似度：如消息发布时间有多接近、有没有共同的话题标签、有没有共同的提及用户等等。另外，

使用哪种聚类方法来完成社交网络的话题聚类任务，也是值得研究的。本文第 3 章中将对这些问题进行详细讨论。

1.3 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。相关研究大致可以分为两类：监督学习技术和筛选排名框架。

谣言检测中关于监督学习技术的研究，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，通过引入已有的或提出新的分类器，对他们进行训练、评测，找出更有效的谣言检测方案。

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] ，而后续的研究陆续提出更有效的特征。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，形成一些新特征。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。Kwon 等人的研究[4]提出了一种基于时间序列分析的特征，用数学模型拟合消息转发数目与时间的关系，并将拟合后估计的 10 个参数作为新特征输入到分类器中。Wu 等人的研究[9]则是提出了一种图核（graph kernel）的度量，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，形成新特征。

而在分类器选择方面以上研究也各不相同，一部分选用了较为简单的分类器，有的研究采用了决策规则[2]、有的研究采用朴素贝叶斯分类器 [5,7]、有的研究采用支持向量机[2,4,9,10]、有的则采用决策树[2,4,7]；也有部分研究选用了较为复杂的分类器：有的采用贝叶斯网络[2,7]、有的采用随机森林[4]、还有的神经网络[7]。以上研究由于使用的数据集不同、特征选取方案也不同，因此表现最好的分类器也各不一样，目前没有定论说哪种分类器最适合谣言检测，这方面仍缺乏理论指导。

另一大类谣言检测框架的相关研究，则是以筛选排名框架为核心。他们的框架不是对每条消息直接进行分类，而是首先对消息数据集进行话题提取，然后对同一话题的消息再抽取统计特征，最后设计出一套可疑度评估方案，筛选出可疑度最高的那些话题，将它们判断为谣言进行输出。

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。

Zhao 等人的工作[11]借鉴以上思路并进行拓展，最终设计出了一套更加完整、健壮的筛选排名框架。该框架分为五步：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，通过与关键词组进行比对将普通消息归到近似的信号消息类中；五，使用监督学习技术对每个消息类进行可疑度排名。该工作论证了此框架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，这两个特点使得该框架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。事实上本文的框架正是在此框架的基础上进行改进和拓展，因此在第 2 章中还会再详细介绍此框架的流程以及与本文方法的比较。

更详尽的相关工作介绍可参考附录 A。

1.4 论文组织结构

本文是基于一个已有的谣言检测框架，对其进行改进和提高，设计和实现一个更加精细、可靠、实用的新框架。本文剩余部分将按照以下结构进行组织：第 2 章将介绍原框架的步骤流程，分析其主要存在的问题，设计改进方案；第 3 章将介绍框架改进中实现的话题聚类技术，提出一种适用于社交网络消息的相关性度量，并用实验说明其有效性；第 4 章将详细介绍各类特征选择技术，并提出的一种新的特征选择方案；第 5 章将简要介绍本框架采用的分类器技术，提出一种新的可疑度排名方法，并用实验说明新的特征选择方案和可疑度排名方法的有效性；第 6 章将进行总结并展望未来的工作。

第2章 方法概述

2.1一个实用的谣言检测框架

本文实现的谣言检测方法基于一项已有工作的谣言筛选排名框架[11]，此框架的实验结果表明它能较早检测出谣言，以及高速处理大规模数据，其实用性很高。因此本文在基本保持其流程框架的基础上，对其进行改进与提高。下面先对原框架的整体流程架构进行介绍，其流程图见图 2.1，检测过程分为五步。

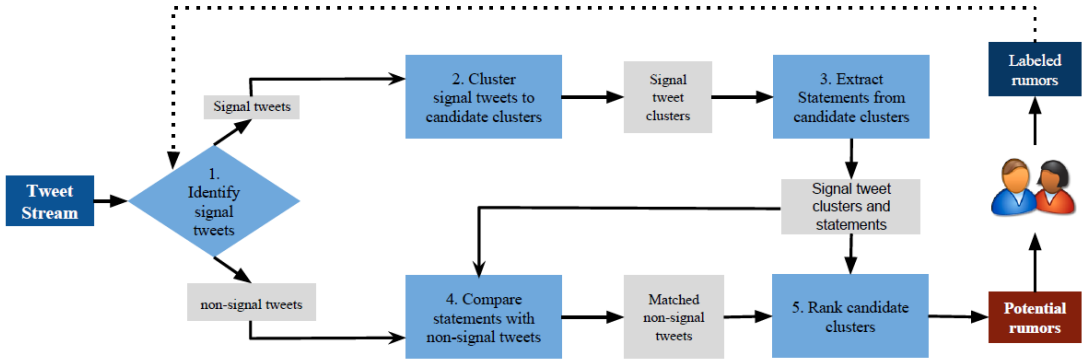

图 2.1 原框架流程图

（图片来源：参考文献中 Zhao 等人的工作[11]）

第一步，信号消息识别。该框架所指的“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，原作者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，对这些消息提取讨论的话题内容，就是潜在的谣言。原框架识别信号消息使用的方法，是文本模式匹配，图 2.2 就是他们使用的匹配模式的正则表达式。由于社交网络消息为短文本，文

Pattern Regular Expression	Type
is (that | this | it) true	Verification
wh[a]*t[?!][?1]*	Verification
(real? | really ? | unconfirmed)	Verification
(rumor | debunk)	Correction
(that | this | it) is not true	Correction

图 2.2 原框架使用的匹配模式

本模式匹配耗时低，即便是对大规模的消息数据集，检测信号消息的速度也非常快。当然，符合模式的消息中有很多谈论的并不是谣言（例如可能只是惊讶某个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。

第二步，信号消息聚类。原框架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。

第三步，对信号消息类提取关键词组集。该框架中的关键词组集（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，非信号消息归类。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，然后遍历所有消息类，计算该集合和该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，否则就再检查下一个消息类是否满足条件；如果所有消息类都不满足条件，则该非信号消息归类失败，不列入任何消息类中。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。原框架选择了决策树作为分类器，它不仅能进行分类，还能输出一个类似“后验概率”的值，表示该消息类是谣言话题的可能性。用训练后的分类器对所有消息类计算此“后验概率”并按此值对消息类从高到低进行排序，找出可疑度排名前的 N 个消息类，作为框架最终检测出来的谣言话题进行输出。

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

2.2 原框架的不足与改进方法

在原框架的文章中用实验说明了该框架处理大数据集的高效率，以及其用于谣言防治的实用性。但该框架仍存在以下两个比较显著的问题：

第一，形成的候选话题重复率过高。原框架为了提高速度，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，因此原框架的每个消息类包含的都是几乎一模一样的消息（例如对同一则消息的大量转发消息），这些消息的文本只差了几个单词。这样的好处是每个类纯度都特别高，可以基本确定类中所有的消息都谈论的是同一话题。但导致的问题是原框架基本无法将表述形式不同，但讨论同一内容的消息聚成一类。如图 1.5 这样的两则消息，尽管它们在讨论同一个话题，但却被原框架分在了两个消息类中。此问题看上去无关紧要，但在实际应用中，并不是检测出谣言候选话题就结束了，而是需要将识别出来的候选消息类交给审查员进行人工复核。这时问题就产生了：审查员反复看到非常多的候选消息类实际上都是在谈论同一个话题，反复审查这些重复的消息类费时费力，而且毫无意义。因此，候选话题重复率过高将大大增加后期人工复核的成本，降低框架的实用性。

第二，检测准确度不高。原框架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，因此后续的监督学习、可疑度排名步骤就尤为重要。但原作者却在这一步只考虑了 13 个比较简单朴素的统计特征，导致了原框架特征的多样性不足。而特征多样性不足，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。因此可以说原框架在可疑度排名这一步还有很大的改进空间，这将是提高谣言检测准确率的关键。

根据原框架以上的不足，本文对其进行了针对性改进，方案如下：

第一，在原框架聚类形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，降低候选消息类的话题重复率。

第二，为候选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以及自动挑选出有效的特征组合，增加框架的检测准确率。

以下对上述方案稍作解释：

1、为什么不直接更换原框架的聚类算法，而是做二次聚类？这是因为原框架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，不尽降低了框架聚类的总时间，而且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的。因此，原框架的每个消息类本身的纯度很高，我们没有必要更换时间代价更大的聚类算法，从头开始对候选消息聚类，而是可以且应该直接在原框架的聚类结果上再进行一轮二次聚类。

2、为什么要在特征上改进框架而不是改进分类器？为什么要引入特征选择技术？这是因为，原框架的特征数目是硬伤。无论是怎样优秀的分类器，如果输入的特征种类太少且不够有效，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。但本文并有没完全不改进分类器，在第 5 章中本文将尝试不同的分类器，并引入一种结合多分类器优点的组合投票技术，对原框架的排名方法进行了一定的改进。回到本段的第二个问题——为什么引入特征选择技术？这是因为目前相关研究提出的特征种类繁多但没有定论说那些特征或特征组合比其它有效；而如果不加思考地将大量特征全部输入分类器，那么其中那些低效、冗余的特征会直接影响检测效果；并且，如果特征数目过多而训练数据集样本量不足够多，很容易造成训练过拟合，大大降低分类器分类准确率。因此，我们不仅要输入足够丰富的特征，还要选择出高效、紧凑的特征子集，通过保证特征的质量来提高分类器的分类排名效果，最终提高框架的检测准确率。而特征选择技术能自动地完成这一任务，所以本文将其引入到了框架当中。

下面的两个章节将分别介绍这两方面的改进：第 3 章介绍框架实现的二次聚类算法与一种适合社交网络话题聚类的相似度度量，第 4 章介绍框架引入的特征和特征选择技术。而在第 5 章将简要介绍新框架采用的分类器，以及一个通过多分类器组合投票的新排名方案。

第 3 章的实验将放在章节 3.4，而第 4 章和第 5 章的实验将统一放在章节 5.4。

第3章 社交网络的话题聚类与消息相似度度量

3.1 框架采用的聚类算法

聚类（clustering）主要是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。由于本框架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，本框架挑选采用了以下几种:

K 均值聚类法[12]（K-means clustering），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德特征空间。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]（kernel K-means clustering），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，本文框架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](hierarchical clustering)，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。本框架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），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。层级聚类法的优势在于其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]（spectral clustering），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（相当于将特征空间降成 n 维），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，都可以尝试用于框架当中，而实际聚类效果如何，还需要实验来评估比较（参考章节 3.4）。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。

3.2 针对社交网络的消息相似度度量

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消息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：

$$sim_{time} = e^{-\frac{\Delta t}{\tau/ln2}}$$

此公式中Δt是两则消息的发布时间间隔，而τ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τ设为不大的值，在本框架的实现中将其设为 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。由于社交网络中的用户在讨论同一个话题时消息中经常会相互提及（@），另外一个用户对另一个用户的消息进行转发时也会带提及符号@，所以如果某消息@另一个用户，而这个被@的用户发布了另一则消息，则这两条消息可能是有关联的，有可能在讨论同一话题；此外在社交网络中，谈论同

一话题的不同消息经常会@相同的用户（比如谈论谣言的消息经常会@谣言事件的当事人，从而提醒他有关于他的谣言），因此@相同用户的两条消息可能彼此也是有关联的，有可能在讨论相同的话题。以上的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。在本框架中采用的用户互动相似度度量定义如下：

$$sim_{itr} = \begin{cases} 1, & if\ two\ tweets\ @\ same\ username \\ & or\ one\ tweet\ @\ author\ of\ another. \\ 0, & otherwise \end{cases}$$

当且仅当两条消息@相同的用户，或者一条消息的作者被另一条消息@，这两条消息的用户互动相似度为 1，否则为 0。可以看出，这里定义的用户互动相似度是二值的，这样设计是因为@关系在社交网络中非常稀疏，一旦两消息出现了以上任何一种@关系，则可以认为他们产生了很高的关联性。在本框架的二次聚类中，不是求两个消息的而是要求两个消息类的用户互动相似度，这里本文采取的做法是，先提取消息类中所有消息的@用户名和作者组成的用户名集合，然后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。其想法来源很朴素：谈论同一话题的消息，其文本内容可能也非常相似。本框架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“#topic”），通过这样用户可将自己的消息归入某类话题，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，它们很有可能是在谈论相同的话题。在本框架实现中，是分别统计两消息类中所有消息的话题标签集合，然后计算这两个两集合间的 Jaccard 相似度作为两消息类的话题标签相似度 $\mathrm{sim_{htg}}$。

命名实体相似度。命名实体（named entity）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]。由于微博上的话题（如谣言）经常为事件，一般会含有事件主体、地点等要素，它们都属于命名实体。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。因此在相似度度量上加入命名实体相关的因素，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），分别提取两个消息类的命名实体集合，然后计算两集合间的 Jaccard 相似度作为两消息类的命名实体相似度 $\mathrm{sim_{NE}}$。

文本单词集的 Jaccard 相似度。其基本想法是：既然谈论同一话题的两条消息的文本可能是相似的，那么其中出现的单词也应该有很高的重叠率。而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确实也非常高；另外即便不是转发，由于两消息的作者可能是看了同一消息来源后进行意见发表，所以它们的措辞和关键词很可能会受到消息来源文本的影响而导致很相似。因此，加入文本单词集合的 Jaccard 相似度，理论上对社交网络的话题聚类会有很大帮助。本文将此相似度的符号记为 $\mathrm{sim}_{\mathrm{jacc}}$。

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），是单词在文本中的出现频率；而第二部分是逆文档频率（inverse document frequency），是含该单词的所有文本的数量占文本集中所有文本总数比例的倒数，它反应的是单词在文本集中的独特程度。此概念被广泛用于文本检索中的相关性排名。当我们计算两消息的文本相似度时，其实并不关心一些停用词（如 is、the、a）的重复率，而是希望求得它们的关键词的重复率；但“关键词”应当如何定义？一种方法是我们可以根据先验知识来定义，比如将话题标签、命名实体取为关键词；而词频-逆文档频率则是将关键词定义为那些文本集中不经常出现的独特单词（因为它们有区分度），在计算文本相似度时通过逆文档频率来调高这些关键词的权重。经过多年的研究与实验发现，词频-逆文档频率相似度的确是很有效的文本相似度度量，所以在社交网络的话题聚类中也推荐引入此相似度。在本框架的实现中，先分别计算出两消息类的词频-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 $\mathrm{sim}_{\mathrm{tf-idf}}$。

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。本框架采用的方式是将这几个相似度加权平均得到综合的加权相似度$\mathrm{sim}_{\mathrm{w}}$：

$$sim_w = w_{time} * sim_{time} + w_{itr} * sim_{itr} + w_{htg} * sim_{htg} + w_{NE} * sim_{NE} + w_{jacc} * sim_{jacc} + w_{tf-idf} * sim_{tf-idf}$$

$$w_{time} + w_{itr} + w_{htg} + w_{NE} + w_{jacc} + w_{tf-idf} = 1$$

上面的公式中，各个 w 符号表示不同相似度的权重，所有的权重之和为 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，在本框架采用的数据集中是加权平均的方式最为有效，因此框架最终实现的、以及实验中汇报的就是这种结合方式。

3.3 实验数据集与聚类评价指标

本小节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中的埃博拉消息数据集，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和抓取，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 章和第 5 章的实验中（章节 5.4 的实验二），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测聚类需要对消息人工地打上真实聚类标签，如果使用完整的数据集则标注工作量太大，所以仅使用其子集。由于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，这个月的消息量适中，因此本文选择了这段时间的消息子集。

实际上，需要标注真实聚类标签的不是消息，因为本章目的是给原谣言检测框架检测出来的谣言候选消息类进行二次聚类，降低候选消息类的话题重复率，所以实际上是要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）。原框架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），原框架聚类后共产生 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 类，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不广的“冷门话题”，这些话题中的消息很少，话题经常只含一个消息类，也就是 686 个类中有很多实际仅有一个元素，这些话题价值不高，即使是谣言也并没有得到广泛传播。

为了去除这些小范围传播的噪声话题，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，原框架聚成的谣言候选消息类共有 214 个，人工标注后其真实话题仅有 67 类，也即候选消息类的话题重复率高达 3.2 倍。换言之原框架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 1.2 和章节 2.2），因此确实有对其实施二次聚类的必要。此删减了“冷门话题”的子数据集称为数据集 B。

表 3.1 数据集 A 与数据集 B 的基本信息

数据集名称	候选消息总数	候选消息类总数	真实的话题总数
数据集 A	9,488	939	686
数据集 B	8,373	214	67

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。

3.4 实验与分析

本文分别在数据集 A 以及数据集 B 上进行实验，对原框架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，新框架二次聚类时使用的相似度度量请参考章节 3.2，新框架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 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（使用高斯核函数），层级聚类使用了 Matlab 的实现版本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 3.2 数据集 A 的聚类结果（指标：NMI）

Method	sim_{time}	sim_{htg}	sim_{NE}	sim_{jacc}	sim_{tf-idf}	sim_{itr}	sim_{w}	sim_{CM}	sim_{g}	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

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

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 3.3 数据集 B 的聚类结果（指标：NMI）

Method	sim_{time}	sim_{htg}	sim_{NE}	sim_{jacc}	sim_{tf-idf}	sim_{itr}	sim_{w}	sim_{CM}	sim_{g}	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）。最上方一行除 Method 的表头均为相似度矩阵的类型，分别是时间相似度（sim_{time}），话题标签相似度（sim_{htg}），命名实体相似度（sim_{NE}），单词集合的 Jaccard 相似度（sim_{jacc}），词频-逆文档频率相似度（sim_{tf-idf}），用户互动相似度（sim_{itr}），将以上 6 种相似度进行加权平均的加权相似度（sim_{w}），两种用于对照实验的基准相似度（sim_{CM}，sim_{g}），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]（sim_{CM}）和贪心法[22]（sim_{g}），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 3.2 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，单独使用时表现最好的是词频-逆文档频率相似度（NMI 均值 0.9436），紧接其后的是单词集合的 Jaccard 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），在采用平均连结或加权连结的层级聚类法时，NMI 更是达到了所有实验结果的最大值 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 种相似度中比基准相似度表现好的为词频-逆文档频率相似度单词、单词集合的 Jaccard 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相似度。

与表 3.2 相比，表 3.3 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。

下面来观察数据集 B 的二次聚类结果（表 3.3）。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（除了简单连结）的聚类效果都很好（平均 NMI 都在 0.7 以上），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 Jaccard 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），说明使用它们单独进行聚类的效果非常不好。同数据集 A 的情况类似，综合考虑这 6 种相似度的加权相似度在所有相似度中聚类表现最好，

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 还高出 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，则能得到所有实验结果中最高的 NMI 值（0.9299），远远超出其它的聚类结果。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。

以上是实验数据观察，下面总结两数据集上实验结果的共性，并作实验分析：

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有很大部分是一个用户看到其它用户的消息后受到启发而发布的，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。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），因为此相似度将定义“关键词”定义为那些区分度高的独特单词，并将其权重调高，从而更准确地判定两文本的相似度。与它相比，单词集合的 Jaccard 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，因此其表现比词频-逆文档频率落后很多。但在章节 3.2 也分析到，单词集合的 Jaccard 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，因此其聚类表现可以也很高，两数据集中到排名第二。时间相似度的聚类表现则是与单词的 Jaccard 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。这是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，所以它也很适合用于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含的元素，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。

对于综合考虑了 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的相似度结合起来，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效果。但问题是，加权相似度的权重配比应该如何决定？本文在实验中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 分配给 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选取出来的参数，有可能会引起过拟合的问题，并不一定适合于其它数据集。那么如果在实际应用中，框架要对一个新来的、没有标注的数据集进行聚类，我们应该使用怎样的权重配比呢？

针对以上提及的过拟合问题，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经常效果不好；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，将大量网格搜索出来的最优权重配比进行平均，获得一个较为“普适”的权重配比。这样的配比不能保证在新数据集上可以获得最优的结果，但由于它是很多个实验最优结果的平均，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”。

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。做法是将数据集 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。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 3.4 各聚类方法在两数据集下最优的权重配比

Mtd-Ds	w_{time}	w_{htg}	w_{NE}	w_{jacc}	w_{tf-idf}	w_{itr}	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；之后每一列（w）表示的是各个相似度矩阵的权重值，每列权重的具体含义可参考章节 3.2 的加权平均公式；最后一列（Total）是每一行各权重之和，由于权重间存在约束，因此都是 1.0。对各聚类方法下的最优权重配比取平均后得到的“普适”配比在最后一行（Avg），可以看到此配比基本符合我们之前实验分析：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 Jaccard 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。比较出乎意料的是，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其实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 Jaccard 相似度获得的比重虽然较大但没有预期那么高，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 Jaccard 相似度的作用，而 Jaccard 相似度由于没有考虑“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 3.5 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 3.5 和表 3.2、表 3.3，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，虽然总体仍不如网格搜索（NMI 均值对比：0.9482 vs 0.9517；0.8480 vs 0.8776），但只有在少数情况下会出现较大差距。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（网格搜索 NMI 值 0.9658，平均评比 NMI 值 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）。

以上实验说明，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。我们之所以要在原框架筛选聚集出来的消息类的基础上，再做一次二次聚类，是因为原框架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，原框架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？首先必须明确，实际上二次聚类后得到的新消息类的总类数是算法预先设置好的：例如 K 均值法和谱聚类法都要在算法运行前设置聚类中心的数目，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。所以直接问二次聚类得到的新消息类有几类没有意义，因为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设置类数是多少呢？又怎么能确定新框架的确能降低话题重复率，并正确聚类？有一种这样的想法：可以通过遍历参数的方法，求出最

高的 NMI 对应的类数，它应该就是最佳类数。这是个有趣的想法，但在解决实际问题中并不能这么做。目前之所以能求 NMI，是因为我们将一个小数据集上的消息进行了聚类标注，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注（实际上我们正是不想去标注才研究聚类算法的）。所以以上遍历参数找最优指标的方法不适用于框架求解真实问题。要想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，人为地尝试几个值，再通过小范围地查看聚类结果来判断预设置的类数是否合理。

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框架二次聚类的效果，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（也即在预设置类数接近真实值时，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 3.4 最后一行），聚类方法选用效果最好的层级聚类，尝试将类数设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 3.6 不同的预定义类数 K 的聚类效果（指标：NMI）

K	K / K_{true}	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 3.6 最左边的一列（K）是算法预设置的类数，第二列（K / K_{true}）是预设置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（67 ±20）时，NMI 指标也都属于可接受的范围或更好。因此可以认为二次聚类能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，框架能将话题重复率压缩到多少。表 3.6 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，框架在聚类效果最好时能将话题重复率平均降到 1.6 左右，而原框架筛选聚类后得出的消息类，其话题重复率是 3.2，也就是说二次聚类将原框架候选消息类的话题重复率降低至 50%左右。即使 1.6 的重复率相比理想的 1.0 仍有一段距离，但可以说本文的改进方案显著改善了原谣言检测框架候选消息类的话题重复率过高的问题。

最后，我们来通过一些例子来说明二次聚类的改进方案的确能将讨论同一话题的消息类聚集在一起，表格如下：

表 3.7 消息举例：新老框架聚类对比

L_{true}	T#	L_{old}	Representative Tweet Text	T#	L_{new}
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?!	1	**3**

			@MinaSayWhat		
		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 3.7 的第一列为真实的话题标签（L_{true}），第二列为每个真实话题的消息数量（T#）；第三列到第五列是原框架筛选聚类生成的消息类的信息，第三列是原消息类标签（L_{old}），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#）；而表格第六列是新框架将原消息类进行二次聚类后得到的新消息类标签（L_{new}）。此例子取自表 3.2 中 H-Avg 行、sim_w 列的实验结果，是新框架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 3.7 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 个原消息类在经过二次聚类后仍没有被正确地聚类，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在特伦顿（Trenton）发现埃博拉病患的谣言，这个话题含有的 3 个原消息类，经过二次聚类都被成功地聚在一起。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（换言之这是一个假的谣言候选话题），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。二次聚类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 Jaccard 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起作用，导致了使用加权相似度的新框架也无法将它们正确聚类。这也正是社交网络话题聚类所面临的挑战：消息文本内容过于杂乱，而相似度计算中的可以给予辅助的因素又经常缺失，导致聚类难度大大增加。

虽然聚类准确率不是 100%，但从表 3.7 可以看出，经过二次聚类，框架能将大部分讨论同一话题的消息类成功聚集在一起，大大降低了候选消息类的话题重复率，进而显著降低后期对谣言候选话题进行复核的人工成本。

至此，原框架候选话题重复率过高的问题得到了很大的改善。而除了话题重复率过高，原框架还面临着一个问题：检测准确率不高。下两章将分别从特征的角度和分类器的角度，尝试对原框架进行改进。

第4章 特征选择技术

4.1 特征选择技术简介

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，因此本文框架仅采用了前两种特征选择技术。以下两个小节将对框架用到的过滤器和包装器特征选择进行概述。

4.2 过滤器特征选择技术

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），本框架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。本节将介绍本框架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，而事实上早期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 Peng 等人的研究[26]，其将评估函数定义为以下形式：

$$E(x_j) = I(x_j, c) - \frac{1}{|S|}\sum_{x_i \in S} I(x_j, x_i)$$

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，x_j 是未被挑选的特征集合中的某个特征变量，x_i 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，这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，可进行互补。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不经常可靠。

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。

Kira 等人的缓和算法[27]（relief algorithm）就是基于样本实例的过滤器特征选择技术。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：

$$w_i = w_i - diff(x_i, near\text{-}hit_i)^2 + diff(x_i, near\text{-}\mathrm{miss}_i)^2$$

其中 w_i 是第 i 个特征的权重，而 x_i 是样本第 i 个特征值，near-hit_i 是近邻命中样例第 i 个特征值，near-miss_i 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，则此特征判定为不适合用于分类，反之如果差距很小说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与类标是相关的；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，那么此特征有可能很适合用于区分不同类别的样例。算法通过遍历大量样本，不断更新每个特征的权重，最后挑选权重最高的前 N 个特征，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。此算法流程很简单，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，因此本框架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不能被满足；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，不同的特征由于范围、分布、绝对值大小不一样等问题，距离函数度量难以统一，但权重迭代公式却是由距离函数直接决定的，最后选择特征则是靠权重高低来进行，如果距离函数对于不同函数本身就不公平，那么算法挑选出来

的特征质量也必然无法得到保证；第四，该算法依然只是考虑了单个特征对分类的影响，如果出现单个特征无法区分样本类别，只有多特征组合才能完成分类的情况，那么此算法将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都没有解决的问题，因为分析特征集整体与类别的关联度并不容易做到——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过程一定程度上就是特征选择的过程，而随着树高度的增长，不同特征组合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。由于过滤器算法独立于分类器：此过程仅涉及特征和类标的数据分析，没有分类器参与（这也是其算法速度快的原因之一），所以选取出来的特征子集不一定适合各种分类器。因此后来出现了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”。

4.3 包装器特征选择技术

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，作为选择的结果。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(2^n-1)个，指数级别的时间复杂度加上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候选的特征子集，最终找到局部最优解。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分类准确度的效益大小。由于整个算法属于全自动化的搜索算法，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）的方法计算评估指标的均值、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。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。该文章通过实验说明，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 更优，才将其设置为新的 b。算法会预先设置一个整数 p，当 b 在 p 轮迭代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。本文框架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 中还将介绍框架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，省去与分类器联系不紧密的相关性分析，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所获得的特征组合，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。

4.4 以过滤器指导起点的浮动式包装器

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？

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。他们用理论和实验说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。因此本文提出以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内进行搜索，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都不会太差的特征组合，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，可以说此方法的确能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。

我们将在章节 5.4 中通过实验对不同的特征选择算法（包括过滤器、包装器、以过滤器指导起点的浮动包装器）选择出来的特征子集对分类器准确度的效益进行评测，说明在原框架中引入特征选择技术有助于框架谣言检测准确率的提高。

4.5 框架特征列表

由于原框架中仅有 13 类特征，其特征多样性不足会成为阻碍框架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特征，将特征扩充成了 45 类。需要注意的是由于本文框架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 新框架使用的特征列表

类型	特征	描述
原框架特征（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

其中原框架相关的特征有 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），这 15 类中除了第一类“信号消息比例”以外其实都是消息相关特征，只是又细分为信号消息的消息特征与所有消息的消息特征两种，以及两者的比值。

新框架中引入的大部分是用户特征，需要特别说明的是以下特征：

发布平均流逝时间：这里流逝时间是指消息发布时间与该消息发布用户的注册时间的差值，因此这实际上也算是消息特征。

用户平均 URL 数量、用户平均简介数量：因为社交网络中用户的个人资料不一定有填写个人主页的 URL 以及个人自述，所以这两个特征对每个用户都是二值的（0 或 1），总体特征是对所有用户特征的平均。

有影响力用户比例：这里所指的影响力是指用户被关注数量与关注数量的比值，比值越高说明其在社交网络中经常扮演消息发布者的角色，越低说明经常扮演的是消息接收者的角色，通过设定一个阈值（本框架设置为 1.0），影响力高于阈值的是有影响力用户，低于阈值的是普通用户。

用户平均转发比：这里是指将每个用户在社交网络中发布的所有消息抽取出来，计算其中转发消息占发布消息总数的比例，最后求所有用户的比例均值。

新框架中也引入了 8 类消息特征，除了第一类中的“消息总数”，其它特征都与消息情感度量相关。其中消息的情感评分、正向情感消息和负向情感消息判断（此外还有中性消息）使用的是由 Tang 等人的研究[32]提出的方法。而消息中情感正向词和负向词判断则是使用了 Liu 等人[33]发布的情感词列表。

最后一类特征是传播特征，其主体是候选消息类中提取出来的所有转发树，也就是那些由转发关系建立的转播树。由于候选消息类含有很多消息，其中包含的转发树可能不止一棵，因此要将所有转发树提取出来以后，计算各转发树的根节点数目、非根节点数目、深度、最大分叉（任意节点子节点数目的最大值），进而计算所有转发树的根节点总数、非根节点总数、最大深度和最大分叉的最大值，形成 4 类传播特征。

至此，框架特征的多样性得到了很大的提高，但是也如本文在章节 2.2 中所述，加入了这么多特征很容易在训练分类器时引起过拟合，导致分类效果低下（章节 5.4 将用实验说明过少的特征与过多的特征都无法使分类器获得良好的分类准确率），因此就需要本章介绍的特征选择技术来自动选取高效、紧凑的特征子集。

下一章将介绍新框架采用的分类器种类，以及引入的多分类器组合投票技术。第 4 章和第 5 章的完整实验与分析将放在章节 5.4。

第5章 分类器与话题可疑度排名

5.1 监督学习技术简介

本文的谣言检测框架的最关键一步是对筛选出来的候选消息类进行可疑度排名，实际上这是一个分类的任务，只是需要分类器能输出代表“可疑度”的后验概率，用以最后的排名。因此这一步需要借助监督学习技术来完成。本小节将对监督学习技术的概念、基本类型进行简要介绍。

监督学习技术，通常被用于分类当中，其要解决的基本问题是：给定一组样本数据和其特征，以及样本的类标签，希望通过找到样本特征空间的一个划分方法，尽可能地将同标签的样本特征点划分在同一个区域，这个过程称之为监督学习中的训练，这些样本称为训练数据；训练后的划分方法可以看作是样本特征空间到标签变量集的一个映射函数，这样的映射函数通常被称为分类器；训练得到的分类器可以接受新样本的特征向量，将其映射成某一类标签，从而完成对新样本类别的评估，也即完成分类任务；实际应用中一个好的分类器并不一定是能将所有类别的训练数据完美地划分开的，而是能够尽可能准确地评估新样本（或称未知样本）的类别的。“监督学习”的命名思想是，分类器训练的过程就像学习一样，而带标签的训练数据就像其老师一样在一旁监督它，确保它的学习质量。

监督学习技术的核心是分类器模型的设计，以及训练方法。不同的监督学习技术这两个要素各不相同。目前比较普遍使用的分类器有决策树、朴素贝叶斯、贝叶斯网络、支持向量机、神经网络等[34]。

5.2 框架采用的分类器

本文实现的谣言检测框架在分类器上采用了决策树与朴素贝叶斯，这两种分类器都不仅能进行分类，还能输出样本属于某一类的后验概率值，用于谣言话题的可疑度排名。以下将简要地介绍这两种分类器。

决策树分类器，是通过一个树型的模型完成分类任务：树中的节点为样本的某一特征，根据特征值的不同，从该节点会延伸出不同的分支，通向不同的子节点；而每一个叶子节点都代表一个类标签。对某样本的分类过程是从根节点开始，根据当前节点的特征种类以及样本该种类特征的特征值决定通往哪一个子节点，然后将通往的子节点作为新的当前节点迭代运行此流程，直到当前节点为某一个

叶子节点，那么该叶节点代表的类标签就是分类器输出的预估类别。对于决策树而言，分类器的训练过程就是建树过程，训练方法即建树策略。决策树的建树过程是自顶而下的，基本流程是挑选特征、生成分支、迭代建树。目前使用最广泛的有 ID3、C4.5、CART 三种决策树，它们的建树过程各不相同。以最早的 ID3 为例，其挑选特征是根据特征对样本分布的信息增益（information gain）大小来进行的；而且不同于其它两种决策树，它没有剪枝策略。决策树可以通过输出叶节点上剩余训练样本的各类别所占比例，作为该叶节点各类别的后验概率。本文使用的决策树是 Matlab 的实现版本①。

而朴素贝叶斯分类器的训练过程，则是利用训练样本中的特征与类别分布关系，估算出样本各类别出现的先验概率，以及给定样本类别后出现各类特征值的后验概率。当需要对一个新样本进行类别预估时，朴素贝叶斯分类器利用训练时估算的各类别的先验概率、特征值的后验概率，通过概率论中的贝叶斯公式，以及所有特征都是相互独立的基本假设，计算出给定新样本的特征值后各类别出现的后验概率，取其中概率最大的类别作为分类器输出的预估类别。朴素贝叶斯分类器的分类过程本身就包括了后验概率的计算过程，因此它能很自然地输出后验概率。本文使用的朴素贝叶斯分类器是 Matlab 的实现版本②。

需要说明的是本框架采用的这两种分类器都是比较简单的分类器，更加复杂的分类器如贝叶斯网络、随机森林、神经网络未被框架尝试，这是未来工作之一。不过根据谣言检测相关研究[1-11]，不是越复杂的分类器就能得到越好的分类效果。

5.3 多分类器投票排名方案

由于不同的分类器各有特色，对于不同的数据集和输入特征，表现最优的分类器也不一样[1-11]。因此研究者、工程人员经常会纠结于选择哪种分类器。通常的做法是将所有分类器都尝试一遍，然后取分类准确率最高的。这不失为一种解决方案，但是能否有更好的做法？既然不同分类器有各自的优势，是否能将这种优势互补，结合多分类器的“智慧”完成分类任务？

答案是肯定的，多分类器组合投票技术就是这么一种方法。其思想是将多个分类器当成多个分类问题的“专家”，让他们各自独立地对新样本进行分类，最后将分类结果进行投票，投票最高的类别作为新样本最终的预估类别。

① http://cn.mathworks.com/help/stats/classificationtree-class.html，Matlab R2014b 版本
② http://cn.mathworks.com/help/stats/classificationnaivebayes-class.html，Matlab R2014b 版本

本文的框架也尝试将此技术引入到话题可疑度排名中，提出一套新的多分类器投票排名方案。具体做法如下：假设要选取前 N 个最疑似谣言的话题，那就让多个分类器选出各自排名前 2N 的可疑话题形成候选列表，然后统计每个话题在各分类器的候选列表中出现的频次，将它们按频次从高到低将其重新排序，最终输出前 N 个话题作为最终的检测结果。

这套新的排名方案做法简单，但设计上有一定的考量：之所以选取排名前 2N 而不是前 N 的话题形成候选列表，是因为每个分类器独立的分类结果不一定准确，前 N 个话题容易混入很多分类器误判的假候选话题，因此只取前 N 不仅会选中不少噪声，还会漏掉很多排名本身靠前但却被假候选话题挤出前 N 的谣言话题，使得参评的真正谣言话题不足 N 个。而通过扩大参评候选话题的范围，能使更多的真谣言话题参与排名，使算法对单个分类器有一定的容错率。而之所以忽略话题在各分类器的排名，直接用出现频次进行排名，是想抹平各分类器对排在最前面那些候选话题的可疑度评估的细微差距，只找出那些被大多数分类器都认为较可疑的话题。而通过这样找出来的候选话题一般是那些特征比较典型的谣言话题，它们不一定能在各分类器的排名中都保持排在最前面，但却能保证被大多数分类器排在中上的位置或更靠前；因此，当一个话题出现在越多分类器的“中上”候选列表中，则它是真谣言的概率越大。相反地，大部分典型的非谣言话题或许会受到了个别分类器的“偏见”，被排在很靠前的位置，但却不大可能被多数分类器都认为“较可疑”而排在较前的位置（除非所有分类器都质量不高）。因此，典型的非谣言话题在所有分类器的“中上”列表的出现总频率不高，通过此方案进行重排名后将在比较靠后的位置，从而不易被最终输出。

下一章节我们将通过实验看到将多分类器组合投票技术引入分类任务和排名任务对提高框架的检测准确率有明显的帮助。

5.4 实验与分析

本章节是第 4 章和第 5 章的实验章节，将全面评测这两章提到的各类特征选择技术、分类器技术以及多分类器组合投票技术对提高框架检测准确率的作用。在章节的最后我们将看到一些框架检测出来的谣言话题消息。

实验一。此实验使用的数据集是章节 3.3 中的数据集 A，是推特埃博拉数据集在 2014 年 11 月的消息子集，经过谣言模式匹配，筛选出 9,488 条谣言候选消息，经原框架聚类后共产生 939 个谣言候选消息类，人工进行聚类标注后得到 686

个类，也就是 939 个消息类中实际有 686 个真实话题；把属于同一话题的消息类聚集在一起，形成 686 个新的候选消息类，成为本实验的样本集。对所有新的谣言候选消息类进行人工进行标注（谣言话题 vs 非谣言话题），得到 182 个谣言消息类，以及 504 个非谣言消息类，形成本实验数据集。实验中对这 686 个候选消息类提取表 4.1 中的 45 种特征，利用第 4 章介绍的所有特征选择技术分别进行特征选择，将挑选出的特征子集再输入分类器进行评测；使用的分类器是章节 5.2 中介绍的决策树与朴素贝叶斯，评测指标有全类别准确度（accuracy）、谣言识别准确率（precision）以及谣言识别 F1 度量（F1-score）三类，评测方法是对数据集 A 进行 10 折交叉验证。

实验细节：实验中过滤器特征选择技术选取的特征个数为预定义参数，通过尝试不同的值汇报最优实验结果；实验中各类包装器特征选择技术的迭代评估方法是在数据集 A 中进行 5 折交叉验证，迭代评估函数分别尝试以上三类评估指标，取最优汇报；章节 4.4 中提出的浮动包装器特征选择技术，其在实验中输入尝试的搜索起点由章节 4.2 中四种过滤器技术的输出提供，同样是汇报最优的实验结果。

实验结果表格如下：

表 5.1 不同特征选择技术在两种分类器中的表现

Classifier	Feature Selection	Accuracy	Precision	F1-score
DT	Original-15	0.6019±0.0462	0.2423±0.1454	0.2079±0.0920
	All-45	0.5978±0.0525	0.2486±0.0973	0.2497±0.0868
	Filter-Pearman	**0.6432±0.0516**	**0.3347±0.1498**	0.3081±0.1063
	Filter-Spearman	0.5787±0.0635	0.2470±0.0996	0.2601±0.0874
	Filter-NMI	0.6222±0.0546	0.3081±0.1280	0.2898±0.0777
	Filter-Relief	0.6178±0.0669	0.2840±0.0891	0.2824±0.0869
	Wrapper-Forward	0.6242±0.0539	0.2856±0.1249	0.2750±0.1007
	Wrapper-Backward	0.6395±0.0682	0.3181±0.0762	**0.3132±0.0893**
	Wrapper-Float	0.6271±0.0826	0.3117±0.1150	0.3047±0.1319
DT-V	/	0.5699±0.0682	0.3127±0.1185	0.3738±0.1177
NB	Original-15	0.6430±0.1243	0.0864±0.1467	0.0617±0.1081
	All-45	0.6741±0.1003	0.3267±0.2932	0.1519±0.1136

	Filter-Pearman	**0.7122±0.0618**	**0.4064±0.1208**	0.2131±0.1208
	Filter-Spearman	0.5842±0.0926	0.3331±0.1186	0.4047±0.1234
	Filter-NMI	0.6866±0.0914	0.3298±0.2692	0.1418±0.0961
	Filter-Relief	0.6393±0.1054	0.2515±0.1350	0.2063±0.1528
	Wrapper-Forward	0.5953±0.0865	0.3522±0.1431	0.3899±0.1294
	Wrapper-Backward	0.6024±0.0942	0.3559±0.1366	0.4089±0.1176
	Wrapper-Float	0.5914±0.1103	0.3578±0.1422	**0.4374±0.1467**
NB-V	/	0.6057±0.1038	0.3695±0.1495	0.4090±0.1462
All-V	/	0.6606±0.0825	0.3896±0.1792	0.3762±0.1396

现对表 5.1 进行说明：

第一列是分类器种类。分别是决策树（DT）和朴素贝叶斯（NB），以及引入多分类器组合投票技术的三种复合分类器：由 6 个决策树共同投票的复合分类器（DT-V），由 6 个朴素贝叶斯共同投票的复合分类器（NB-V），以及由 6 个决策树加上 6 个朴素贝叶斯共同投票的复合分类器（All-V）。同一种模型的不同分类器实例（如 6 个决策树），挑选的是各类特征选择技术生成的最优分类器。

第二列是特征选择种类。其中对照组有：取原框架相关的 15 类特征（Original-15），取新框架所有的特征共 45 类（All-45）。实验组有：过滤器选择技术共 4 种，前 3 种是基于相关度分析的过滤器，其评估函数分别是皮尔曼相关系数绝对值（Filter-Pearman）、斯皮尔曼相关系数绝对值（Filter-Spearman）、归一化互信息（Filter-NMI），最后 1 种是基于样本实例的缓和算法过滤器（Filter-Relief）；包装器选择技术共 3 种，分别是前向搜索模式（Wrapper-Forward）、后向搜索模式（Wrapper-Backward），还有以过滤器为起点指导的浮动搜索模式（Wrapper-Float）。另外，所有复合分类器是由多分类器结合而成的，这些子分类器使用的特征选择技术各不相同，于是三个复合分类器的特征选择一栏均打上了“/”符号。

接下来的第三到第五列。分别是三类评估指标，实验结果是 10 折交叉验证得到的 10 组评测指标的统计值，按“平均值±标准差”的形式给出。

需要说明，评估指标有三种，但这三种指标互有关联，仅有一类指标很高并不能说明分类器较优。例如，拥有很高的谣言准确率，但谣言 F1 度量低，说明谣言检测的召回率很低，框架并不实用；再如，拥有很高的全类别准确度，但谣言 F1 度量低，说明分类器仅对非谣言话题判断良好，对谣言的判断却很差，作为一

个谣言检测框架的检测器，也并不实用。

那么，我们分析时应该怎么看这三类指标呢？本文推荐的顺序是先 F1，再谣言准确率，最后全类别准确度。因为作为谣言检测框架，不仅想测准谣言，还想尽可能测出更多谣言，因此 F1 度量是最重要的指标；然而，由于框架最终要利用分类器排名取前 N 个，而分类器的谣言识别准确率是其直接影响因素，因此它不能太低；而全类别准确度参考价值最低，因为数据集中非谣言数目远高于谣言数目（504 vs 182），假如分类器将所有消息判断为谣言，其值高达 0.73（高于所有的实验结果），因此仅是此值高并没有意义；但在 F1 度量和谣言准确率较高时，参考全类别准确度可以看出分类器对非谣言的识别情况，这时此值如果过低，则可能会有很多非谣言被误认为谣言。

通过观察实验结果可得到以下分析结论：

引入更多特征和特征选择技术有必要性。实验结果验证了本文之前的论述：原框架的特征数目过少、多样性不足将导致检测准确率偏低。无论是哪种分类器，使用原框架的 15 类特征得到的谣言识别 F1 度量都很低（0.2 与 0.06 左右），类似地，谣言识别准确率也很低，这说明其对谣言的识别能力非常低下。同样，如果不进行特征选择，将所有 45 类特征输入两种分类器，得到的结果也不好（F1 度量为 0.25 和 0.15）。虽然相比 15 类原框架特征，全特征的确能让分类器得到提高，但效果不大。然而在引入了各种过滤器、包装器的特征选择技术后，在几乎所有情况下分类器的表现都是有明显提高的，特别是在朴素贝叶斯分类器中提高幅度特别显著。以上结果说明了向原框架引入更多特征与特征选择技术的必要性。

过滤器特征选择技术有不稳定性。观察四类过滤器的实验结果，发现表现参差不齐：在决策树分类器中，归一化互信息和缓和方法的表现中等，斯皮尔曼的表现是所有选择技术中最差的，但皮尔曼的表现却是所有选择技术中最好的（三个指标中两个排名第一，一个排名第二），甚至压过了迭代优化的包装器。但是在朴素贝叶斯分类器中，皮尔曼的表现却不能说最好，虽然它的全类别准确度和谣言识别准确率仍然排名第一，但 F1 度量却仅排名第五，而且绝对数值仅有 0.25，远低于最高的 0.43，这说明此分类器召回率很低，仅能检测出极少的谣言，即便准确率高也没有用。从实验结果看，归一化互信息和缓和方法过滤器选出的特征都不适用于朴素贝叶斯分类器，其 F1 度量处于选择技术的最后两名，归一化互信息的选择甚至还不如全 45 类特征不作选择的表现；但在决策树中表现最差的斯皮尔曼却在朴素贝叶斯中表现最好（F1 高达 0.40），甚至与包装器技术接近。以上实验结果说明过滤器特征选择技术选出的特征子集表现并不稳定，有时表现

极差有时却极好，因为相关度分析和基于实例的近邻分析并不总是可靠。从平均来说，过滤器技术表现中等，经常处于包装器技术之下；但不可否认的是过滤器能快速地选出较为有效的特征子集，对分类器识别准确率有一定的提高。

包装器特征选择技术有卓越性和稳定性。观察实验中 3 种包装器技术的评测数据，发现其整体表现比过滤器技术要卓越很多，特别是在朴素贝叶斯分类器上。这是因为包装器是经过长时间的迭代搜索加实测评估后，挑出其认为最优的特征组合，这样选出的特征子集是大量候选中的局部最优，自然比较稳定。可以看到包装器选出的基本所有特征子集都有很高的表现，除了决策树上的前向搜索模式，其表现略低于大部分过滤器，这是因为算法在此次迭代中并没有达到最终的收敛，运行时间是其它包装器的 20 倍以上，猜想是搜索进入了一些较深但终点全局不优的分支；由于在达到局部最优前算法就被作者强制终止了，所以其表现较差。而在实验运行的数十次包装器算法中，这是唯一不收敛的例子。以上实验说明，包装器技术能大幅提高分类器的表现，大多数情况下搜索结果都很可靠，此类技术具有卓越性与稳定性。

以过滤器指导起点的浮动包装器有一定的潜力。实验结果中浮动包装器在朴素贝叶斯分类器中的表现是很好的，在保持最高的 F1 度量（0.43）的同时，其谣言准确率也排在第二位（0.35），而谣言准确率排在第一的是 F1 度量很低、无实用性的皮尔曼过滤器，因此可以说在此分类器中，浮动包装器的结果是所有特征选择技术中最好的。这是因为浮动包装器以过滤器选择出来的多个特征子集作为尝试的搜索起点，而且其子集变换操作丰富，这些因素使得浮动包装器不易过早地落入全局不优的局部最优当中。但是浮动包装器也并不一定能搜索出全局最优的解，如决策树中分类器中，其 F1 度量仅排名第三，在反向搜索包装器与皮尔曼过滤器之后（虽然与它们差距不大）。这是因为启发式搜索并不是用理论指导能完全控制的，虽然浮动包装器能一定程度保证其搜索结果全局较优，但搜索起点很大程度决定了搜索结果的优劣，即便尝试了过滤器选择出来的多个搜索起点，也不能一定会比单个起点的结果要好（如反向包装器）；另外，浮动过滤器的搜索起点中涵盖了皮尔曼过滤器的特征选择结果，但最终的实验指标却不如皮尔曼过滤器，这是因为包装器在迭代中使用的评估方法（或数据集）与测试或实际中使用的评估方法（或真实数据集）有一定的差距，包装器在评估过程中认为有比皮尔曼过滤器选出的特征子集更优的子集，于是淘汰了前者，但不意味着在测试实验或真实数据中此优劣关系也成立。这是所有包装器都会遇到的现象：其认为的局部最优在切换数据集后不一定是局部最优的。以上实验分析说明，浮动过滤器通

过增加搜索宽度的确增加了搜索到全局更优结果的潜力，但这不是必然的。

对比两种分类器，决策树更稳定，朴素贝叶斯最好表现更优。从实验数据就能看出来决策树在任何特征子集下，其表现都不会太差，即便是原框架的 15 类特征，或是全部的 45 类特征，其 F1 度量也有 0.20 和 0.24；而且各特征选择的表现相差不大，波动范围在 0.26 到 0.31 间。决策树分类器的稳定性是与它的模型原理分不开的，由于建树过程本身就是一种特征选择的过程，因此无论输入多差的特征子集，决策树都能挑出其中较优的特征放在最上层；而下层则通过剪枝把无用的特征自动忽略了。相比之下，朴素贝叶斯分类器的波动就非常大，在 15 类特征和 45 类特征下，其 F1 度量都极低；而在各特征选择技术下，最差的表现 F1 度量能低至 0.14，远低于决策树的平均范围，但最优的表现 F1 度量到达了 0.43，远高于决策树能达到的最高表现。这是因为朴素贝叶斯是一种很依赖于特征质量的算法：它的独立性假设，导致了输入的所有特征都具有平等性，最后每个特征都是作为连乘的一项汇总到后验概率中；这时如果输入的特征都与类别关联性不高，则概率估算必然不准；如果特征子集中有不少优秀特征，但同时又混有大量噪声特征，那么后者就会给后验概率公式连乘上大量无用的噪声项，从而干扰优秀特征准确的概率估算。因此，当特征选择结果不佳时，朴素贝叶斯通常表现也非常糟糕；但是当特征选择结果优秀时，统计学与概率论的模型就能帮助其获得卓越的分类效果。以上实验说明决策树分类器在不同特征下更加稳定，但朴素贝叶斯配合良好的特征选择技术则能达到更高的表现。

多分类器组合投票技术能抹平分类器的偏见，其分类准确率超出子分类器平均水平很多，且能接近或超越最优的子分类器。观察多分类器的实验结果可以发现，6 决策树复合分类器的表现远超于实验中任何一种特征选择下的决策树分类器（尽管复合分类器是它们组成的）；6 朴素贝叶斯复合分类器的表现虽然没有超越排名第一的子分类器，但也高于第二的子分类器，而且也远高于子分类器的平均表现；至于 6+6 的复合分类器，其表现优于 6 决策树复合分类器（特别是谣言准确率有很大提高），略逊于 6 朴素贝叶斯复合分类器，但也高出所有子分类器的平均水平很多。在判断每个消息话题的类别时，复合分类器会先询问各分类器的意见，只有当认为是谣言的票数超过一定的比例，才将话题判断为谣言；这样的做法能够将分类器的偏见抹平，只取那些比较统一的意见，有利于检测出那些较为典型的谣言——因为它们即便被较差的分类器判断错，也能被大多数分类器判断出来；而因为典型的谣言一般占谣言话题的大多数，所以当子分类器中仅有为数不多的噪声分类器时，复合分类器让优秀的分类器优势互补，将准确率提高到

平均水平以上很多，甚至超越每一个子分类器。至于那些意见不统一的非典型谣言，就不是复合分类器所擅长的：一般只有极少数优秀分类器能将它们判断正确，而且不同分类器擅长判断的非典型谣言不同，因此这种谣言的判断通常比较看运气，可能会将少量优秀分类器的正确意见打压下去，这就是为什么有时复合分类器会略为不如某些极优秀的子分类器。

以上就是实验一的全部分析。

实验一中使用的数据集很小，仅取了一个月的数据，不到完整数据集的十分之一；其评测方法是通过交叉验证求平均分类准确率，这遵循了评估分类器的传统方法，却与实际的谣言检测框架不同——实际的检测框架是通过可疑度排名取前 N 的方式来确定最终输出的谣言话题，理论上即便整体分类准确率不高，只要谣言后验概率排在前面的那些话题中真正谣言的比例高，框架就是成功的。因此，本文设计了更为贴近真实场景和框架真实应用流程的实验二。

实验二。此实验使用的数据集是完整的推特埃博拉数据集，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。经过框架前几步的模式匹配和聚类，共筛选出 293,686 条谣言候选消息，共 13,974 个谣言候选消息类。对这些候选谣言消息类提取表 4.1 中的 45 类特征，根据实验一中训练得到的不同分类器，输入相应的特征，然后使用分类器对这 13,974 个候选消息类进行可疑度评估（计算他们属于谣言的后验概率），然后进行排序，将可疑度最高的前 20、50、100 的候选消息类挑选出来，进行人工复核，看其中真正的谣言话题数量，计算检测框架的 Top-N 准确率。实验结果的表格如下：

表 5.2 不同特征选择与分类器组合下框架的谣言检测准确率

Classifier	Feature Selection	Top-20		Top-50		Top-100	
		R #	Precision	R #	Precision	R #	Precision
DT	Original-15	9	0.45	19	0.38	35	0.35
	All-45	9	0.45	23	0.46	32	0.32
	Filter-Correlation	12	0.60	24	0.48	36	0.36
	Filter-NMI	9	0.45	21	0.42	38	0.38
	Filter-Relief	8	0.40	19	0.38	40	0.40
	Wrapper-Forward	**13**	**0.65**	18	0.36	41	0.41
	Wrapper-Backward	11	0.55	26	0.52	39	0.39

	Wrapper-Float	12	0.60	**30**	**0.60**	**48**	**0.48**
DT-V	/	15	0.75	28	0.56	51	0.51
NB	Original-15	6	0.30	12	0.24	28	0.28
	All-45	10	0.50	16	0.32	30	0.30
	Filter-Correlation	10	0.50	**24**	**0.48**	46	0.46
	Filter-NMI	11	0.55	17	0.34	28	0.28
	Filter-Relief	3	0.15	9	0.18	23	0.23
	Wrapper-Forward	7	0.35	**24**	**0.48**	**54**	**0.54**
	Wrapper-Backward	**12**	**0.60**	23	0.46	44	0.44
	Wrapper-Float	**12**	**0.60**	**28**	**0.56**	49	0.49
NB-V	/	13	0.65	31	0.62	57	0.57
All-V	/	18	0.90	32	0.64	57	0.57

对表 5.2 进行说明：第一列是分类器种类，其符号意义与表 5.1 的符号完全相同，不再赘述；第二列是特征选择，大部分符号意义也与之前相同，只有符号“Filter-Correlation”之前没有出现，它表示以皮尔曼相关系数绝对值以及以斯皮尔曼相关系数绝对值为评估函数的过滤器特征选择技术，将它们选择的特征子集取最优作为此项的特征选择结果；从第三列开始就是实验结果，“Top-N”表示经谣言检测框架可疑度排名后，可疑度最高的前 N 个候选话题，“R#”表示其中真正的谣言话题的个数，而“Precision”表示谣言检测框架的 Top-N 检测准确率。

从表中的数据可以得到与实验一非常相符的结论：

仅是原框架的15类特征，或使用新框架全部45类特征的检测准确率并不高。在丰富特征种类的基础上引入各类特征选择技术明显提高了框架的表现，特别是在朴素贝叶斯分类器下更为显著，甚至能将准确率提高到仅用原框架 15 类特征的准确率的 2 倍。这也说明了引入更多特征和引入特征选择技术的有效性。

过滤器选择技术在决策树下表现良好，仅略低于包装器选择技术一点，这与实验一数据是相符的，是因为决策树模型对特征选择有鲁棒性；而在朴素贝叶斯分类器中，过滤器技术的不稳定性就体现出来了，在决策树中 Top-100 准确率最差的相关系数过滤器，在朴素贝叶斯中却表现极好，Top-100 准确率甚至比反向搜索的包装器还高（虽然远低于其它两种包装器），而另外两种过滤器技术在朴素贝叶斯中却表现极差，它们的 Top-100 准确率甚至不如原框架的 15 类特征以及全

45 类特征。此实验再次说明过滤器选择技术的有效性与不稳定性。

而包装器选择技术的卓越性和稳定性在此实验中再次得到佐证。在两种分类器中，包装器的表现都优于过滤器，尤其在朴素贝叶斯中差距更为明显；而且无论是哪种搜索模式，包装器选择出来的特征表现都非常好，发挥非常稳定。

至于以过滤器指导起点的浮动包装器，在实验二中更体现了其潜力。决策树下，其 Top-20、Top-50 和 Top-100 指标都在所有特征选择技术的最高水平之列，其中 Top-50 和 Top-100 准确率不仅排名第一，而且领先其他选择技术很多。在朴素贝叶斯下，浮动包装器的 Top-20 和 Top-50 准确率综合水平远比其它两种包装器高，Top-100 准确率高于反向包装器但低于正向包装器，但仍是一个很高的值。这再次说明本文提出的浮动包装器搜索到全局更优的可能性更大，虽然不是必然。

关于分类器的比较，观察 Top-100 准确率很容易就能得出与实验一相同的结论：决策树更稳定，朴素贝叶斯波动大，但配合良好的特征选择技术其检测准确率能达到比决策树高不少。

对于多分类器组合投票技术的优越性，实验二能比实验一更能说明。其中 6 决策树复合分类器的 Top-100 准确率比任何一种特征选择生成的决策树都要高，而且比除了浮动包装器外的所有决策树还高很多；对于在决策树下表现极好的浮动包装器，复合分类器虽然在 Top-50 指标上略低，但 Top-20 和 Top-100 准确率都比它更优。再来看看 6 朴素贝叶斯的复合分类器，其表现在 Top-20、Top-50 和 Top-100 指标上都压过所有贝叶斯分类器很多，其中 Top-100 准确率甚至比所有决策树分类器与 6 决策树复合分类器都高很多。至于 6+6 复合分类器（All-V），作为集合了两类共 12 个分类器智慧的复合分类器，其表现没有像实验一中那样被较差的子分类器稍微拉低，而是成为整个实验二中最好的分类器，虽然 Top-100 准确率只是与 6 朴素贝叶斯并列，但 Top-20 和 Top-50 指标是绝对的第一，而且其 Top-20 准确率高达 0.9。因此实验二的结果很好地说明了多分类器组合投票技术对框架在真实应用场景下的谣言检测准确率，有卓越的提高作用（其原因的理论分析以及具体的投票方法描述在章节 5.3 的最后）。

一个有趣的现象是，三种复合分类器的 Top-20 和 Top-50 指标都非常高，特别是 Top-20 准确率，6 朴素贝叶斯复合分类器在所有实验数据中排名第三（0.65），6 决策树分类器虽然 Top-100 准确率排名不在前三，但 Top-20 却在所有实验数据中排名第二（0.75），而 6+6 复合分类器的 Top-20 不仅排名第一，而且数值很高（0.90）。作者分析后认为这并不是巧合，而是复合分类器能改变候选话题的谣言分布，使得排名最靠前的候选话题含有更多真正的谣言。本文提出的复合分类器

的投票重排名方法描述可参见章节 5.3 最后。此方法倾向于将那些在各分类器中都排名较前的候选话题重新排在候选列表最前面，这样的候选话题不一定在各分类器中排名都是最前的，但它们是各分类器都“同意”为较疑似谣言的话题，这种话题一般而言就是那些特征典型的谣言话题，它们能被大多数分类器识别出来并给出较高的可疑度。在复合分类器的重排名中，看重的是“共同意见”，于是排在最前面的那些话题基本上就是众分类器共同的“靶子”，自然谣言的比率高。所以实验中才会出现复合分类器 Top-100 指标不一定最高，但 Top-20 准确率都很高的现象；而且复合分类器中含越多子分类器，这个现象就更明显，甚至能让 Top-20 准确率高达 0.9。

复合分类器的以上特点是实用性非常高的：在真实场景下消息数据集规模更为巨大，进行人工复核的审查员没有时间去看很多消息，因此可疑度排名最前的那些候选话题的谣言检测准确率就尤为重要，需要其中尽可能多地出现谣言，减轻审查员的工作负担；而复核分类器的投票重排名技术的特点正是能够完成这一任务，因此引入多分类器组合投票技术，能大大提高谣言检测框架的 Top-N 准确率，从而显著提高框架实用性。

经过改进后的谣言检测框架的 Top-100 准确率为 0.57，比原框架的 0.35 提高了 63%；而其中改进后的 Top-20 准确率为 0.90，比原框架的 0.45 提高了 100%。以上实验结果表明，改进方案对于提高谣言检测准确率有明显的成效。

以上就是实验二的全部分析。

在实验一和实验二中我们看到的更多是数值化的实验评估数据，没有看到框架实际检测出来的谣言，这就如同“隔靴挠痒”。因此在本章节的最后，我们来看一下改进后新的谣言检测框架检测出来的谣言消息都有些怎样的话题，同时也看看框架误判为谣言的普通话题是如何的。请看下面两表：

表 5.3 新框架检测出的前 20 个谣言话题

Representative Tweet of Rumor Topic
#Rumors of Ebola cases in M'sia not true, says Health Ministry - The Sun Daily http://t.co/0Na48CSmgR
DOH: Stop joking, spreading rumors about Ebola in PH https://t.co/2jtWlqoMBh | https://t.co/fqwWUA9wHa http://t.co/JNB1hFsb2m
RT @MiIeyCyrusBish: BREAKING NEWS: There is a rumor of a Scary Outbreak of Ebola through chocolate. Causes Scare over Halloween.
Crazy Rumors Circulating That Michael Essien Has Ebola!!! http://t.co/Q7rFBzoekD http://t.co/2RQr9zQERB

‘Why Jesus, why?’: CNN ‘exclusive’ about Ebola in hair extensions makes rumor rounds: You’d t... http://t.co/ILhajTKeFs via @TwitchyTeam
RT @Naija_Genius: Actor Van Vicker has rubbish rumors of him contracting Ebola in Liberia http://t.co/KWo0GP3iJI @metronaija
The Ogun State Government has debunked claims that the dreaded Ebola virus is in the state http://t.co/HaN5Wxj9gs
Life Ebola Caused by Red Cross Shots? More Conspiracy Theories Emerge: A rumor is going around saying that Ebola… http://t.co/FLRnNklpnc
"@adoringlikeari: ARIANA MISSED A CHARITY EVENT IN NYC BECAUSE SHE WAS SCARED OF GETTING EBOLA OMFHXGHDH" RUMOR RUMOR RUMOR
US Ebola Conspiracy? Biological Warfare in US Rumors Emerge; Roundup: There’s been rumors and conspiracy theories... http://t.co/tdY0Vjp6d6
#ebola Ebola funding rumored to be in funding bill http://t.co/VX7UHTWaG5
What??? "@UberFacts: There are people in Sierra Leone, Africa that think Ebola is a hoax created by doctors to steal blood from patients."
4chan is trying to spread some shit rumor about ebola. If you see the hashtag "#ebolaindoritos", it's horseshit. http://t.co/PSriKtBaoX
What!? Zombie Rabies-Ebola Hybrid Virus in the Making? 100% Proof of Patented Vaccine- I Don't Make This Stuff Up... http://t.co/CBFo3UeyMn
Akon Crowd-Surfs in Giant Bubble at Concert in Africa, Spurring Rumors That He Was Avoiding Contracting Ebola http://t.co/xcQrcqXGVV
Really? @UberFacts: A second outbreak of Ebola has begun in the Congo, independently of the West Africa outbreak."
For Real? Pastor Kumuyi Heals Ebola Victim Through A Phone Call http://t.co/wdTg7DRLyy Via @SkelewuTv
RT @JeffreyGuterman: BREAKING NEWS [unconfirmed]: Boyfriend of #Dallas nurse with #Ebola reportedly admitted to hopsital with Ebola symptoms h…
What? RT @OroyoEubanks: CDC Says Ebola Vaccine Only Works on White People http://t.co/3LZ0nzqWHJ
Ebola crisis rekindles concerns about SECRET research in Russian military labs. Whaaaaaat?!... http://t.co/MR2e0udste

表 5.4 新框架误判的前 10 个非谣言话题

Representative Tweet of Non-Rumor Topic
The Internet is already full of false rumors about Ebola: http://t.co/bCUXBQTpTj
“@PerezHilton: @AZEALIABANKS I think you and Ebola would get along real well!” Ebola really? You couldn’t think of anything else to say?
“@CP24: BREAKING: Two patients are in isolation at a London,Ontario hospital for possible exposure to the Ebola virus.” WHAT?!?!?!? ??
What?!? RT “@bbcworldservice: A town full of orphans – the Liberian village where every

mother has died of Ebola
So just how contagious is Ebola, really? http://t.co/HLAKNfozon
RT @MzBeeezSoSassii: Hot spots? Really? ?? RT @cnnbrk Canada will stop accepting visa applications from Ebola hot spots, official says.
From AIDS to Ebola: In rumor control, only the tech changes http://t.co/8HnBkoa7tm
How Do You Get Ebola, Really?: Amid continued confusion over how Ebola spreads, the World Health Organization … http://t.co/h8mDezNzXk
but… RT @Chunchi: RT @Neauxp: WHAT? RT"@HuffingtonPost: MORE: Liberia to prosecute man who brought Ebola to US http://t.co/0V6XfyhSRC"
RT @CDCemergency: @Ryan_2ka 1) Learn how ebola spreads 2) counter rumors with facts and 3) give aid if you can: http://t.co/iyPjOfODH4

从表 5.3 中看到，框架检测出的谣言话题范围广阔：有关于谣传体育明星感染埃博拉病毒的话题，有关于某地区出现埃博拉病例的话题，有埃博拉是美国虚构出来的阴谋的谣言，还有称埃博拉疫苗只对白人有效的谣言。大部分谣言话题检测出的候选消息都带有强烈的惊讶、不确定、怀疑、质疑感情色彩的单词，或者其本身就是辟谣性质的消息，这些正是此框架用以检测的线索信号，模式匹配筛选的那一步就是要找出这样的信号消息。这是基于一个假设：任何的谣言在传播过程中都出现一部分明智的用户会发出这些质疑的声音。

然而表 5.4 中那些被错误检测出的非谣言话题通常也带有这样的信号词，其中有大部分是惊讶语气词，而且带有很多问号和叹号加强惊讶的感情色彩，但实际上消息惊讶的对象不是谣言，而是一些轰动性的新闻或不当的言论；而有些消息带有单词“really”，但却不是表示惊讶而是用作副词，大意为“真正地”；另一部分的消息中带有关键词“rumor”（谣言），但却是说当前谣言满天、需要进行谣言控制等等的言论，并不是在谈论具体的谣言。从这里我们就能看出其实原框架通过模式匹配检测出谣言消息的同时，也混入了很多难以区分的噪声候选消息。

即便如此，本章节的两个实验也充分说明加入了更多特征和引入特征选择技术的新框架相较原框架的谣言检测准确率有显著的提高；而本文提出的浮动包装器技术以及引入的多分类器组合投票技术能进一步提高检测框架的检测准确率，增加框架表现的稳定性，以及框架在真实应用场景下的实用性。

第6章 总结与展望

6.1 总结

本文实现了一个完整的社交网络中的谣言检测框架，此框架基于一个已有工作的谣言检测框架。通过分析，本文提出原框架存在的两方面问题：候选话题重复率过高，以及特征种类太少导致的检测准确率不高。接下来本文对这两方面的问题分别提出解决方案，并用实验说明方案的有效性。

对于第一个问题，本文的改进方案是在原框架聚类出话题的基础上再做一次聚类。对此，本文选择了基于相似度矩阵的聚类算法，并讨论了 6 种适合社交网络话题聚类的相似度度量，考虑了包括时间、用户互动和文本内容等因素，然后提出用加权平均的方式结合这 6 种相似度度量。本文用实验说明了加权相似度对社交网络话题聚类的卓越效果，并讨论了权重配比方案。框架尝试了 4 类不同的聚类算法，本文从理论和实验上说明了层级聚类法在社交网络话题聚类问题中的优势。本文还用实验说明了加入二次聚类后，新框架能在大幅降低候选话题重复率的同时保持较高的聚类效果，很大程度改善原框架的第一个问题。

对于第二个问题，本文提出应引入更多新类型的特征，并通过特征选择技术自动选择出优秀的特征子集来提高分类器的谣言检测率。本文引入了过滤器、包装器两种特征选择技术，并提出了一种将两者优势结合起来的新方法——以过滤器指导起点的浮动包装器技术。框架采用了决策树与朴素贝叶斯两种分类器，并用实验对比了不同特征选择技术与不同分类器组合对框架检测准确率的提高作用，分析说明了各种特征选择技术以及分类器的特点，并说明了以过滤器指导起点的浮动包装器的搜索稳定性和特征选择效果的卓越性。此外，本文引入了多分类器组合排名技术，提出了一种多分类器投票重排名的方案改进原框架的排名器。理论和实验都说明此方案对提高框架的 Top-N 检测准确率有卓越效果，而且此方案能提高框架在真实场景下的实用性，显著改善原框架的第二个问题。

经过以上两方面的改进提高后，本文实现的新谣言检测框架相比原框架，其候选话题重复率更低、谣言检测准确率更高、在真实场景下更实用。

6.2 展望

经过本文的改进，原谣言检测框架已经有很大的提高。但作者在实验过程中

也发现新框架仍有很大的改进空间。如果未来还想继续对其进行提高，作者认为可以在以下几方面进行努力：

一，改进模式匹配方案。原框架筛选出谣言候选消息的方案，是通过一些线索信号，也即怀疑、质疑、否定的单词或句式，通过文本模式匹配来找到信号消息，再通过信号消息找到更多谣言候选消息。但在实验中作者发现原框架使用的部分匹配模式会引入很多噪声候选消息，使得框架在筛选的第一步就加重了后续的检测负担。由于本文仅在原框架的聚类方案和分类排名方案上做了改进，未对模式匹配方案进行更改，因此这方面仍有很大的提升空间。

二，降低聚类的时间成本。由于改进方案中二次聚类使用了基于相似度矩阵的聚类方法，而相似度矩阵的计算的时间复杂度是 $O(n^2)$的，表现较优的层级聚类法的时间复杂度是 $O(n^3)$的，因此在大规模数据下，二次聚类的耗时过长，这方面仍可以做改进。目前想到的方案是将时间进行分段，仅对每段时间内的候选消息进行聚类，从而降低总体的聚类时间。

三，继续引进监督学习技术。由于本文仅尝试了决策树和朴素贝叶斯两种分类器，因此还能通过尝试引入更复杂的分类器模型如贝叶斯网络、神经网络等，来提高框架的谣言检测准确率。

插图索引

表格索引

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.

[32] Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). Learning sentiment specific word embedding for twitter sentiment classification. Proceedings of the 52nd ACL, 1555-1565.

[33] Liu, B., Hu, M., & Cheng, J. (2005). Opinion Observer: Analyzing and Comparing Opinions on the Web. Proceedings of the 14th WWW, 342-351.

[34] Bishop, C. M. (2006). Pattern Recognition and Machine Learning (Information Science and Statistics). Springer.

致　　谢

感谢刘世霞老师对我论文的指导，以及毕设期间对我的莫大帮助。感谢王希廷学姐给我提供实验的数据集，以及经常解答我在框架实现和撰写论文中的疑惑。感谢刘梦尘学长给予我宝贵的文章修改意见。感谢同实验室的肖建楠同学在毕设期间跟我同甘共进，相互鼓励完成答辩。感谢刘世霞老师组内的每一人在我大学的科研过程中对我的支持，感谢这个大家庭。

感谢父母对我从小的培养，以及大学期间对我的支持和照顾。感谢我的大学室友赵雷彧、周伯威、钱珺、俞则明、张家政、包煜、周晔共同营造的良好宿舍环境，让我能专心学习和科研，最终顺利完成毕设。感谢我的女朋友柏彤对我的陪伴、照顾与宽容，在我毕设最困难的时期鼓励我渡过难关。

感谢微软亚洲研究院 IEG 组的各位成员，包括王超、刘旭东、罗文斌、冯旦旦、郭旭、王君、何琨，特别感谢我的实习导师陈刚，他们在我实习期间给予了很大帮助，让我在这个组感觉到家一般的温暖，并在我毕设最忙的时候顶替我的工作，给予我足够的时间和空间完成论文。

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：钟仰新　　日　期：2016.6.12

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框架，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。Cao 等人在他们的文章中[1]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。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框架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]。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。

Seo 等人在他们的研究中[6]就是采取了这样的方式。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有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有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。

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[3]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。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 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。

Qazvinian 等人的研究[5]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”。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分类器，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。

Yang 等人的研究[10]借鉴了 Qazvinian 等人的研究，但研究的对象从 Twitter 转移到了新浪微博。他们借鉴并拓展了 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的客户端发布的，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。

Sun 等人的研究对象也是新浪微博中的谣言[7]，他们借鉴了 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。在 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%。

Castillo 等人在他们的工作中对 Twitter 数据设计了两个分类任务[2]，一是消息是否具有新闻价值，二是消息是否可信。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。

Kwon 等人的研究[4]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。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。

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，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 $O(n^4i)$，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 0.9 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每一对微博都进行相似度计算，也就是 $O(N^2)$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、用户、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框架

谣言检测还有一小部分研究提出的解决方案是筛选排名框架。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。

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 是他们所设计的框架，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

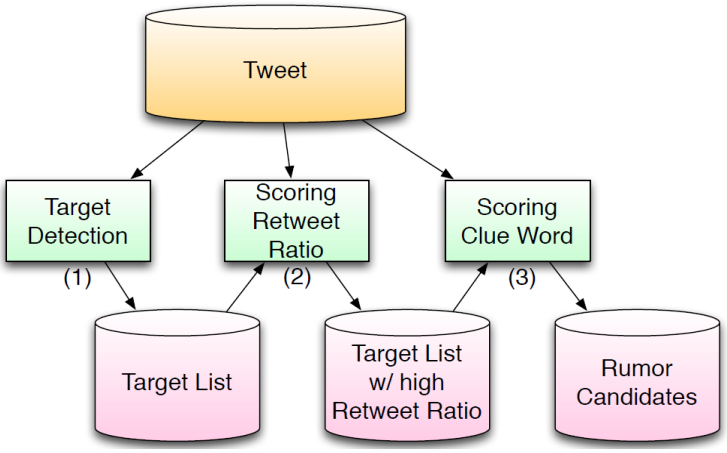

图 A-1　Takahashi 等人的框架

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。

此研究虽然提出了一套筛选框架，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，整个框架首先经过“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、规模性很高的框架。

Zhao 等人的研究[11]设计出的筛选排名框架具有高规模性，并可以用于谣言的早期检测。图 2 是他们的框架，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

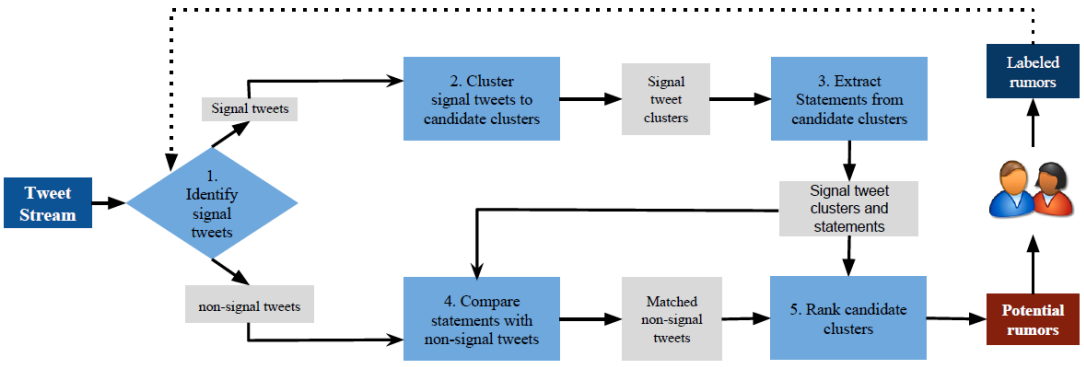

图 A-2　Zhao 等人的框架

Pattern Regular Expression	Type
is (that | this | it) true	Verification
wh[a]*t[?!][?1]*	Verification
(real? | really ? | unconfirmed)	Verification
(rumor | debunk)	Correction
(that | this | it) is not true	Correction

图 A-3　信号 tweets 检测采用的模式规则

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。

值得注意的是此研究中的第五步用到了有监督学习技术，这表明此框架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，该框架能从每天产生的 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，此框架能对 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；由于该框架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），所以此框架可用于谣言的早期检测（此结论经过了实验说明）。最后，由于框架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（该框架处理每天产生 tweets 的 10%只需要 30min）。

当然，目前该框架使用的聚类算法、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，谣言检测中的筛选框架一般采取的技术都很简易，框架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框架。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；而筛选排名框架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框架的设计思路，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	钟仰新	学号	2012013297	班级	软件 21
论文题目	社交网络中的谣言检测				
主要内容以及进度安排	社交网络中存在大量谣言，而此类平台中的用户、消息数量巨大，这使得人工审核的成本高昂。因此，谣言的自动检测技术拥有重大的理论研究和实际应用意义。本论文计划基于最新的谣言检测框架，对其两方面不足进行改进：1. 其候选话题重复率过高，计划对候选进行二次聚类，并研究适合社交网络话题聚类的相似度度量；2. 其谣言检测准确率较低，计划加入更多谣言检测常用的特征，并引入特征选择技术进行改进，计划对特征选择技术进行创新，以及改进原框架的可疑度排名方案，提高 Top-N 谣言检测准确率。 进度安排：中期答辩以前完成原框架的全部实现，完成聚类算法和相似度度量的研究与实验；中期答辩后一个月，完成对原框架特征的拓展以及特征选择技术的引入，改进特征选择的方法并进行对比实验；最终答辩以前完成可疑度排名方案的改进和相应的对比实验，完成论文的撰写和修改，并准备好答辩的材料。 **指导教师签字：** 刘世霞 **考核组组长签字：** 张枫 2016年 3 月 16 日				
中期考核意见	该同学制定的工作计划合理，目前进度符合工作计划，建议继续按照工作计划完成课题内容。 **考核组组长签字：** 张枫 2016 年 4 月 20 日				

指导教师评语	本文围绕社交网络中的谣言检测问题，展开了谣言检测技术的探索和实现工作，对一个已有的谣言检测技术存在的两方面问题（候选话题重复率过高、谣言检测准确率过低）进行改进，设计实现了一个更精确、实用的谣言检测方法。其主要贡献是：1. 探讨了 6 种适合社交网络话题聚类的相似度度量，并用加权平均的方式将它们结合考虑，得到一个更有效的度量；2. 提出一种更有效的特征选择方法，名为“以过滤器指导起点的浮动式包装器”；3. 提出一种有效的、基于多分类器投票思想的可疑度排名方案。论文内容完整、条例清晰、重点突出。论文工作量饱满，是一篇优秀的本科论文。 **指导教师签字：** 刘世霞 2016年 6 月8 日
评阅教师评语	该论文研究社交网络上的谣言检测方法，论文首先对现有框架进行系统、准确的分析，在此基础上通过提出新的相似度度量方法，特征选择方法和多分类器融合的排名机制，在多个指标上实现对传统框架的大幅超越。该论文选题新颖，且具有较高实用价值，算法设计与实现体现了作者对研究问题的深入分析和创新性思维，建议组织答辩并建议评优。 **评阅教师签字：** 张枫 2016 年 6 月 8 日
答辩小组评语	该论文选题符合专业培养方案，论文内容充分详实，较好实现了课题目标，建议通过论文答辩。 **答辩小组组长签字：** 张枫 2016 年 6 月12 日

总成绩： 94

教学负责人签字： 张慧

2016 年 6 月14 日