基于大数据计算框架的分布式新闻聚类系统设计

卢献华[1] 王洪俊[2]

(北京信息科技大学 北京 100101)[1] (北京拓尔思信息技术股份有限公司 北京 100101)[2]

摘 要 对海量的互联网新闻进行快速热点聚类是一个重要的研究方向。针对大规模文本聚类的几个关键问题(相似度计算、分布式聚类、聚类结果概要生成),文中设计并实现了一个基于 Spark 计算框架的分布式新闻聚类系统。该系统采用 GPU 加速的深度相似度算法进行新闻文本的相似度计算,得到新闻之间的相似关系,并采用图聚类算法进行新闻聚类,最后采用标题压缩技术形成热点描述,生成最终的聚类结果。实验结果证明,文中提出的系统具有较高的执行效率和良好的可扩展性,可以有效地处理大规模新闻的热点聚类任务。

关键词 分布式图聚类,深度相似度计算,GPU 加速,标题压缩,大数据

中图法分类号 TP3 **文献标识码** A

Design of Distributed News Clustering System Based on Big Data Computing Framework

LU Xian-hua[1] WANG Hong-jun[2]

(Beijing Information Science and Technology University, Beijing 100101, China)[1]

(Beijing TRS Information Technology Co., Ltd., Beijing 100101, China)[2]

Abstract Rapid clustering of massive Internet news to generate hot topic is an important research direction. Aiming at several key problems of large-scale text clustering: similarity calculation, distributed clustering and clustering result summary generation, this paper designed and implemented a Spark-based distributed news clustering system. Firstly, the GPU-accelerated deep similarity algorithm is used to calculate the similarity relationship of news texts. Then the graph clustering algorithm is used for news clustering. Finally, a short title for each class is generated as the class description. Experiments show that the proposed system has high performance and good scalability, and can effectively handle hotspot clustering tasks of large-scale news.

Keywords Distributed graph clustering, Depth similarity calculation, GPU acceleration, Title compression, Big data

1 引言

随着互联网和信息技术的飞速发展,如何从海量的新闻中发现网络热点已经成为一个重要的研究方向,并且其在网络舆情监控、信息安全、市场分析等领域有着广泛的应用前景。

针对大规模文本聚类的几个关键问题(高性能文本相似度算法、分布式聚类框架、聚类结果概要生成),文中设计并实现了一套针对海量新闻的分布式图聚类系统,该系统支持大数据计算框架 Spark,首先采用深度文本相似度计算方法,该方法可以充分利用 GPU 的高算力,实现高速计算,构建新闻之间的相似关系,然后采用基于 Spark[1]/GraphX[2] 的连通图聚类算法进行分布式聚类,最后采用标题压缩技术形成热点描述,并输出最终的聚类结果。实验结果表明,该分布式聚类系统具有较高的执行性能和良好的可扩展性,可以有效处理大规模新闻的热点聚类任务。

2 研究现状

2.1 文本相似度计算

在进行聚类前,需要先计算新闻文本之间的相似度,根据相似度把内容相近的新闻聚合到一起。文本相似度是表示两个或多个文本之间匹配程度的一个度量值,相似度越大,说明文件相似程度越高,反之文件相似程度就越低。

针对文本相似度计算,研究者们提出了很多算法,如向量空间模型(Vector Space Model)、概率模型 BM25、统计语言模型等[3-4]。传统的相似度模型使用的文档向量是一个高维向量,每个维度对应一个词语,各词语间相互独立,无法捕捉不同词语间的语义关系。为了实现快速相似度计算,须对词语建立倒排索引,以实现高效计算。

近年来,基于深度神经网络的自然语言处理研究取得了快速发展,在很多语言处理任务上都取得了优于传统方法的性能,包括语义相似度计算。Le 等[5]提出了一种无监督学习方法,可以获得句子、段落和文档的分布式向量表示。该方法习得的向量是固定维度的稠密向量,可使用欧氏距离等算法计算句子、文档之间的相似度。Kusner 等提出了一种无监督的基于词向量的计算文档距离的方法:Word Mover's Distance(WMD)[6]。该算法利用 word2vec 的特性,将文档表示为基于词嵌入的加权点云,两个文档 A 和 B 之间的距离定义为 A 中所有的词移动精确匹配到文档 B 中点云的最小累积距离。Kiros 等提出了一种有监督的句子编码器模型 Skip-Thoughts[7],学习对句子的语义属性进行编码,将输入的句子编码成固定维度的稠密向量表示,可以用于语义相关性、句子

卢献华(1964—),女,硕士,主要研究方向为大数据、软件工程、项目管理。

情感分类等任务。为了解决短文本聚类中词汇个数少、描述信息弱导致维度高、特征稀疏和噪声干扰等特点，孙昭颖等[8]提出了一种基于深度学习卷积神经网络的短文本聚类算法。梁吉业等[9]提出了一种面向短文本分析的分布式表示模型，将词对主题模型的主题信息融入 Paragraph Vector 中，不仅在模型训练过程中利用全局语料库的信息，而且还利用 BTM 显性的主题表示完善 Paragraph Vector 隐性的空间向量。

固定维度的稠密向量表示方法有一个优点，可以应用 GPU 加速算法来实现更高性能的相似度计算。这个优势是传统稀疏向量还不具备的。

因此，本文采用了基于深度学习的文本相似度计算方法，将文本表示成稠密向量形式。基于 GPU 的相似度加速算法采用了 Facebook 开源的 fasiss 高性能相似度计算库[10]。

2.2　分布式计算框架

经过文本的相似度计算，可以得到新闻之间的相似关系，然后可以采用聚类技术，将存在内容相似关系的新闻聚合到一起，生成新闻热点。

目前可用的聚类算法大多仅适于处理小规模的数据，在当今信息爆炸的背景下，互联网新闻数量呈指数级增长，文本特征空间的维度也急剧增大，这都会严重减弱聚类算法的类别划分能力，同时也大大延长了算法的运行时间，因此如何对大规模文本数据进行快速、有效的并行聚类计算成为了一个很有价值的研究方向。

MPI(Message Passing Interface)等传统的并行计算方法存在开发复杂、扩展性不好等问题，已无法满足日益增长的互联网大规模数据处理的要求。而 MapReduce 作为一种新的并行处理模型，具备可扩展性高、应用方便等特点，得到了广泛的关注与应用[11-12]。陈德华等[13]提出了一种高效的分布式并行图聚类算法，在传统的 MapReduce 基础上，对传统的 MR_LSH 算法的并行化进行改造，使其可以在分布式的集群环境中实现对大规模图数据的高效聚类。刘鹏等[14]提出了一种基于 Spark 的 k-means 文本聚类并行化算法，利用 RDD 编程模型充分满足了 k-means 频繁迭代运算的需求。

本文采用了基于 Spark/GraphX 的连通图聚类算法。Spark 是加州伯克利分校开源的通用分布式框架，该框架使用内存保存中间输出结果，减少了磁盘读写开销，因此更适用于数据挖掘与机器学习等需要迭代的 MapReduce 算法，可用来构建大型的、低延迟的数据分析应用程序。GraphX 是 Spark 上的一个专门用于图计算的弹性分布式图系统，通过 Spark 将数据并行和图并行有效地结合在一起，将大规模的图数据集按照一定规则分布到集群的各个结点上，然后在各个结点上处理一部分数据集，在计算过程中只涉及结点间的网络通信，不需要磁盘 I/O 开销，且可以根据数据集的规模任意添加机器，可扩展性好，极大地提升了图计算的存储和计算效率。

2.3　短标题生成

完成新闻热点聚类后，需要给出每个类别的内容描述，以便用户更好地理解聚类结果。

通常，可以从聚类结果中选取代表性新闻的标题作为类别描述。由于中文新闻标题较长，一般有二三十个字，为了在有限的网页空间中显示更多的聚类结果，需要在不影响语义的基础上对标题进行压缩改写，缩短标题长度。

关于标题生成的研究方向如下：

(1)基于规则的方法。这类方法利用基于手工语言的规则来检测或压缩文档中的重要部分。Dorr 等提出了一种基于手工编写规则的标题压缩算法 Hedge Trimmer[15]。基于规则的方法简单轻巧，但无法获取文本中的复杂语义关系。

(2)基于统计的方法。这类方法利用统计模型来学习标题和文档中词语之间的相关性。Witbrock 等[16]提出了一种基于统计方法的标题生成算法，包括从文档中选择标题词和将标题词排序组织成可读文本序列两个阶段，该方法可以有效地生成给定长度的新闻短标题。

(3)基于摘要的方法。标题可以被视为非常简短的摘要。Filippova 等提出了一种基于词图的最短路径的多句压缩方法[17]。杨冰等[18]提出了一种融入显著性事件信息的标题生成模型，首先利用互增强原则学习显著性事件，并指导生成候选语句，然后根据候选语句构造词图，再结合路径显著性、流畅度以及覆盖度等因素，设计相应的排名策略以生成最终的标题。

(4)基于深度学习的方法。Sun 等[19]提出了一种基于深度学习生成模型的方法，利用 pointer-network 机制的 copy 和 generate 两种机制，来解决商品标题在移动设备上展示过长、需要压缩标题的问题。

本文提出了一种短标题生成算法，对每个类别的所有新闻标题进行压缩筛选，生成代表性的短标题作为该类的描述信息。

3　系统设计与实现

3.1　总体设计

本文设计的分布式聚类系统的核心是一个基于 Spark 的分布式集群，该集群由一个主节点(master)和数个子节点(worker)组成，具体如图 1 所示。

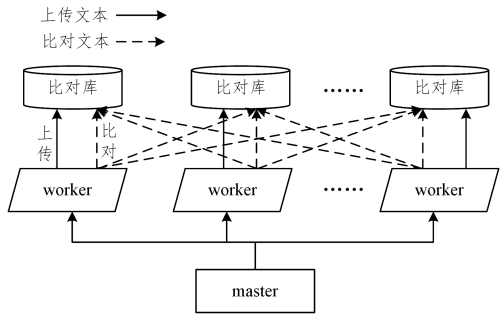

图 1　分布式新闻聚类系统

(1)在执行分布式聚类任务时，主节点对将进行聚类的文本数据集合 T 进行编号，每篇文本对应一个唯一的 id。

(2)主节点将文本集合 T 拆分为数块，并将其平均分发至每个子节点。每个节点上的文本集合记为 T_i，其中 i 为节点序号。

(3)在每个子节点部署了用来计算文本相似程度的比对库，每个子节点在接收到聚类数据 T_i 后，先将 T_i 中的所有文本和其 id 上传至比对库中，作为后续比对的样例文本。

(4)每个子节点将 T_i 中的每篇文本作为比对文本，向所有的子节点上的比对库发送相似度计算请求，获得与该文本相似的所有文本的 id 集合，并根据相似文本对的 id 生成文本相似关系集合$\{(a,b)\}$，其中(a,b)表示 id 为 a 的文本和 id 为

b 的文本具有相似关系。

(5)以文本 id 为节点,相似关系为边,构建基于文本相似关系的拓扑图。对图进行连通图聚类,将图中的每一个连通图作为一个类簇,连通图中的节点 id 即为该类中所属文本的 id。

(6)对每个类别的内容生成短句形式的短标题描述,十个字左右,方便用户理解聚类结果。

3.2 基于 GPU 的相似度计算算法

在进行聚类前,需要先计算新闻文本之间的相似度。与传统相似度计算方法不同,本文采用了基于深度学习的文本相似度计算方法。首先对新闻语料进行分词,使用开源的 word2vec 工具训练一个词嵌入模型(word embedding),再对新闻文本进行分词,使用训练好的词嵌入模型计算文本中所有词语的词嵌入(word embedding)的平均值,将每篇新闻文本表示成稠密向量形式,最后计算稠密向量之间的欧氏距离相似度。考虑到稠密向量可以使用 GPU 加速,采用了 Facebook 开源的 fasis 高性能相似度计算库。对分配到每个子节点上的聚类数据 T_i 进行处理,将其表示成稠密向量形式,统一加载到 fasis 库中,然后根据接收到的相似度计算请求,对库中的稠密向量进行快速的相似度计算,从而得到最终的相似度结果。

由于目前的稠密向量算法在短文本相似度上的效果更好,本文采用了 3 组稠密向量:基于标题的稠密向量、基于首段的稠密向量、基于全文的稠密向量。对于一篇新闻,分别按 3 种类型的文本生成向量,并计算相似度,然后进行相似度加权,取平均值作为最终的相似度。

3.3 图聚类算法

将得到的相似度结果表示成(文档 a,文档 b,相似度)的形式,以文本 id 为节点,相似关系为边,可以生成基于文本相似关系的拓扑图,对图进行基于 GraphX 的连通图聚类。

具体方法为识别图中所有连通图,过滤掉节点数和边数低于某个阈值的连通图。将图中剩余的符合条件的每个连通图作为一个类簇,连通图中节点 id 即为该类中所属文本的 id。连通图聚类算法的一个缺点是两个内部关系紧密的类簇可能由于一个相似度不是很高的关系被聚合到一起,从而形成一个关系较为松散的类簇。为了提升聚类效果,计算每个类簇的聚合度,对于那些聚合度不高的类,去除其中相似度较低的关系并重新进行图聚类。

聚合度计算方法如下:由文中采用的聚类方法可知,每一个类簇都是一个连通图,用该连通图中所包含的边的数量除以图中包含的节点数,即可获得该类的聚合度。

3.4 短标题生成

完成新闻的图聚类后,需要对每个类别的内容生成短标题描述,方便用户理解聚类结果。具体过程如下:

(1)标题串拼接

首先对每个类的标题集合进行拼接处理,将每条新闻的标题作为一个句子拼在一起,形成一段长文本。其中,每个标题的结尾用句号分隔,每个标题内部如果有空格,则需要用句号代替。这样做的目的是把每个标题分成独立的句子,内部的小标题也分成一个个独立的句子。

(2)高频串提取

提取标题长文本中的高频字串,并统计其出现频率。这里的高频字串指的是长度超过 5 个汉字或单词的字串,且该字串在文本中至少出现两次。高频串的提取方法是统计 N 元字串,把满足以上条件的高频词串作为候选单元,在候选单元产生过程中,不仅要滤掉低频统计词串,还要滤掉同频子串。

所谓同频子串指,如果串 A 包含串 B,且串 B 和串 A 在文中出现的次数相同,则称串 B 是串 A 的同频子串。例如"中国"和"中国人民"在文中都出现了 4 次,且"中国人民"包含"中国",则"中国"是"中国人民"的同频子串。

(3)高频串过滤排序

对得到的高频串集合进行筛选,方式如下:

1)判断该词串在长文本中是否曾经出现在句首,如果没有则过滤掉该串;

2)判断该词串在长文本中是否曾经出现在句尾,如果没有出现则过滤掉该串;

3)判断该词串中是否包含标点符号,如果包含,则过滤掉该串;

4)对剩下的词串按在长文本中的出现频率从高到低的次序进行排序;

5)选择出现频率最高的词串作为最终的短标题。

图 2 是本系统的一个可视化展示效果图。图中的不同色块代表不同的聚类结果,色块中的文字就是该类的短标题生成结果。

图 2 可视化展示效果

4 实验过程及结果分析

为了验证分布式聚类算法的有效性,在实际系统上进行了测试。

4.1 实验数据集

实验选取从互联网中文网站下载的 2017 年 3 月 10 日—2017 年 3 月 20 日的新闻网页,共计 100 多万篇。为了验证不同数据量下的聚类性能,分别选取 37 万篇和 102 万篇两个数据集。

4.2 实验环境

系统配置为:3 台 Linux 服务器,每台的配置为双核处理器,64 GB 内存,GPU 卡 GTX1080 Ti;服务器上部署了 Apache Spark 系统和文中设计的分布式海量文本聚类系统。为了验证本系统的可扩展性,设计了单台服务器和 3 台服务器的对比实验。

4.3 实验结果

1)深度相似度计算方法与向量空间模型的相似度性能比较如表 1 所列。

表 1　性能比较

配置	数据量(102 万条)	相似度计算耗时/min
102 万数据，3 台服务器	VSM	256
	基于 CPU 的深度相似度	112
	基于 GPU 的深度相似度	77

2)不同数据量的聚类测试对比结果如表 2 所列。

表 2　对比结果

数据量/条	GPU 相似度计算耗时/min	图聚类耗时/min	总时间/min
37 万	28	5.7	34
102 万	77	21	98

3)单台服务器和 3 台服务器的对比实验结果如表 3 所列。

表 3　对比实验

相似度方法	数据量/条	单服务器耗时/min	3 台服务器/min
基于 GPU 的深度相似度计算	37 万	95	34
基于 GPU 的深度相似度计算	102 万	290	98

从测试结果可以看出：

1)基于深度学习的相似度算法经过 GPU 加速后，性能有了较大的提升，但由于相似度计算部分还包括特征向量生成等部分，因此整体性能的加速效果只提升了一倍左右，节省了相似度计算阶段 50%的时间。

2)从单节点和多节点的对比实验可以看出，本文设计的分布式架构具有较好的可扩展性，通过增加节点即可有效地支撑更大的数据量。

总体而言，本系统可以快速地对百万级新闻文本进行聚类，具有较高的执行性能。

结束语　针对互联网海量新闻的热点发现问题，本文设计并实现了一个分布式新闻聚类系统，该系统在开源大数据计算框架 Spark 的基础上，应用了深度文本相似度计算、分布式图聚类、标题压缩等技术，可以有效地对采集的海量网络新闻进行快速的聚类分析和热点归纳，帮助用户及时了解网络热点动态。

实验证明，本文的系统具有良好的处理性能和可扩展性，足以支撑百万件新闻文献的快速聚类。

参 考 文 献

[1] Apache Spark™- Unified Analytics Engine for Big Data [EB/OL]. http://spark.apache.org/.

[2] GraphX | Apache Spark[EB/OL]. http://spark.apache.org/graphx/.

[3] ROBERTSON S,ZARAGOZA H. The Probabilistic Relevance Framework:BM25 and Beyond[J]. Foundations and Trends® in Information Retrieval,2009,3(4):333-389.

[4] PONTE J M,BRUCE C W. A language modeling approach to information retrieval[J]. Research and Development in Information Retrieval,1998:275-281.

[5] LE Q,MIKOLOV T. Distributed representations of sentences and documents[C]// Proceedings of The 31st International Conference on Machine Learning (ICML 2014). 2014:1188-1196.

[6] KUSNER M,SUN Y,KOLKIN N,et al. From Word Embeddings To Document Distances[C]// Proceedings of the 32nd International Conference on Machine Learning(2015). 2015:957-966.

[7] KIROS R,ZHU Y K,SALAKHUTDINOV R,et al. Raquel Urtasun and Sanja Fidler. Skip-Thought Vectors[C]// NIPS,2015. Curran Associates,Inc,2015:3294-3302.

[8] 孙昭颖，刘功申. 面向短文本的神经网络聚类算法研究[J]. 计算机科学，2018，45(S1)：392-395.

[9] 梁吉业，乔洁，曹付元，等. 面向短文本分析的分布式表示模型[J]. 计算机研究与发展，2018，55(8)：37-46.

[10] faiss：A library for efficient similarity search and clustering of dense vectors[EB/OL]. https://github.com/facebookresearch/faiss/.

[11] 海沫. 大数据聚类算法综述[J]. 计算机科学，2016，43(S1)：380-383.

[12] 李建江，崔健，王聃，等. MapReduce 并行编程模型研究综述[J]. 电子学报，2011，39(11).

[13] 刘鹏，滕家雨，丁恩杰，等. 基于 Spark 的大规模文本 k-means 并行聚类算法[J]. 中文信息学报，2017(4)：150-158.

[14] 陈德华，解维，李悦. 面向大规模图数据的分布式并行聚类算法研究[J]. 计算机研究与发展，2012(suppl 49)：222-227.

[15] DORR B,ZAJIC D,SCHWARTZ R. Hedge trimmer:a parse-andtrim approach to headline generation[C]// Proceedings of the HLT-NAACL 03 on Text Summarization Workshop, Stroudsburg. PA, USA: Association for Computational Linguistics, 2003:1-8.

[16] WITBROCK M,MITTAL V. Ultra-Summarization:A Statistical Approach to Generating Highly Condensed Non-Extractive Summaries[C]// Proceedings of SIGIR 99. Berkeley: ACM, 1999:315-316.

[17] FILIPPOVA K. Multi-sentence compression:Finding shortest paths in word graphs[C]// Proceedings of the 23rd International Conference on Computational Linguistics. Stroudsburg, PA, USA:Association for Computational Linguistics,2010:322-330.

[18] 杨冰，孙锐，姬东鸿. 融入显著性事件信息的标题生成方法[J]. 计算机工程与应用，2016(24)：236-240.

[19] SUN F,JIANG P,SUN H,et al. Multi-Source Pointer Network for Product Title Summarization[C]// Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM2018). Torino:ACM,2018:7-16.