基于半监督学习的中文社交文本事件聚类方法*

郭恒睿[1]，王中卿[1]，李培峰[12*]，朱巧明[12]
[1]苏州大学计算机科学与技术学院，苏州，中国
[2]苏州大学人工智能研究院，苏州，中国
hengruig@outlook.com, {wangzq, pfli, qmzhu}@suda.edu.cn

摘要

面向社交媒体的事件聚类旨在根据事件特征对短文本聚类。目前，事件聚类模型主要分为无监督模型和有监督模型。无监督模型聚类效果较差，有监督模型依赖大量标注数据。基于此，本文提出了一种半监督事件聚类模型(SemiEC)，该模型在小规模标注数据的基础上，利用LSTM表征事件，利用线性模型计算文本相似度，进行增量聚类，利用增量聚类产生的标注数据对模型再训练，结束后对不确定样本再聚类。实验表明，SemiEC的性能相比其他模型均有所提高。

关键词： 社交媒体事件聚类；增量聚类；文本相似度

Semi-supervised Method to Cluster Chinese Events on Social Streams

Hengrui Guo[1], Zhongqing Wang[1], Peifeng Li[12*], Qiaoming Zhu[12]
[1]School of Computer Science and Technology, Soochow University, Suzhou, China
[2]AI Research Institute, Soochow University, Suzhou, China
hengruig@outlook.com, {wangzq, pfli, qmzhu}@suda.edu.cn

Abstract

Event clustering on social streams aims to cluster short texts according to event contents. Event clustering models can be divided into unsupervised learning or supervised learning at present. The unsupervised models suffer from poor performance, while the supervised models require lots of labeling data. To address the above issues, this paper proposes a semi-supervised incremental event clustering model SemiEC based on a small-scale annotated dataset. This model encodes the events by LSTM and calculates text similarity by a linear model, and then clusters short texts on social streams. In particular, it uses the samples generated by incremental clustering to retrain the model and redistribute the uncertain samples. Experimental results show that this model SemiEC outperforms the traditional clustering algorithms.

Keywords: event clustering on social media , incremental clustering , text similarity

基金项目：国家自然科学基金(61772354，61836007)，国家自然科学基金青年基金项目(61806137)，江苏高校优势学科建设工程资助项目。

第十九届中国计算语言学大会论文集，第634页-第644页，海南，中国，2020年10月30日至11月1日。

1 引言

在如今的网络时代，随着移动互联网的发展，信息交互变得前所未有的简便快捷。QQ、微信、微博、抖音、快手等社交媒体深入走进了人们的生活，改变了人们的生活习惯。研究表明，社交媒体对于新事件的反应要比传统媒体更加敏锐(Petrović et al., 2010)。因此，对社交媒体中的文本进行数据分析有着非常重要的意义。其中，事件聚类是社交媒体中事件检测的重要步骤(Aggarwal and Subbian, 2012)。

事件聚类旨在根据文本事件特征的不同对社交媒体中的文本进行聚类。社交媒体中多为短文本，且文本内容具有多样性、随意性，包含较多的干扰词。传统的无监督聚类模型难以准确提取社交文本的事件特征，因而得到的事件聚类结果一般准确度较低。Wang and Zhang (2017)采用了有监督的深度神经网络模型对社交文本进行聚类，增强了聚类效果，但面对海量的社交文本，该方法需要大量的文本标注工作。

基于此，本文提出了一种半监督的中文社交文本增量事件聚类模型SemiEC(Semi-supervised Chinese incremental Event Clustering model)。SemiEC模型利用LSTM(Hochreiter and Schmidhuber, 1997)提取文本特征，利用线性模型计算两个文本属于同一事件的概率，在此基础上进行增量聚类。该模型利用增量聚类过程产生的标注样本对模型进行再训练。在无需额外数据标注工作的同时，帮助模型学习更多的事件信息，提高聚类效果。同时，对于聚类过程中的不确定样本，暂时不进行聚类，在结束后用再训练后的模型进行重新聚类，可以防止较差样本影响簇心表征，影响模型再训练，提高对不确定样本的聚类准确度。实验表明，SemiEC模型与基准模型相比在各项聚类指标上均得到了提高。

2 相关工作

目前，大部分事件聚类研究主要基于词的特征。与长文本不同，短文本聚类存在高维稀疏的问题(Aggarwal and Subbian, 2012)，因此一些学者考虑引入外部特征。其中Mathioudakis and Koudas (2010)以及Saeed et al. (2019)利用突发性关键词来预测短文本的重要性，并通过这些重要的短文本来检测事件进行聚类。Nguyen and Jung (2015)通过考虑事件的发布时间、扩散程度和扩散敏感性，采用时间特征来检测事件，并进行聚类。除此之外，Li et al. (2012)探索了用户在社交媒体数据中的影响，利用文本内容特征、用户特征和使用特征来检测事件并进行聚类。Mcminn and Jose (2015)借助了文本的命名实体特征来加强事件检测的效果。

为了解决传统方法高维稀疏的问题，Cai et al. (2005)通过局部保存索引(LPI)将高维文本投影到低维语意空间，同时使语意相关的文本在低维空间中也彼此接近。Qimin et al. (2015)使用Kmeans聚类算法对特征词集进行聚类得到特征簇，用特征簇表示句向量，从而解决了向量空间模型维度爆炸的问题，同时提高了聚类效果。Zhou et al. (2018)以word2vec词向量为基础，结合时序关系，提出了JS-ID'F顺序来进行文本嵌入。Arora et al. (2019)提出了对文本的SIF Embedding，通过对词向量进行加权平均，再用PCA和SVD对其进行一些修改，得到文本的低维向量表示。Xu et al. (2015)(Xu et al., 2017)采用了基于DCNN的深度神经网络学习文本的深度特征表示。该模型首先通过现有无监督降维方法得到文本的二进制编码，然后将文本通过Word Embedding输入卷积神经网络，将文本的二进制编码作为模型的训练目标。将卷积层与输出层之间的中间特征向量作为文本的深度特征表示，是一种基于自训练的无监督模型。

对于社交媒体中的流式数据，常用的聚类算法有singlepass增量算法和局部敏感哈希(LSH)算法。对于到来的新样本，singlepass聚类算法首先需要计算新来文本与已有事件的相似度，若相似度超过阈值，则将其加入相似度最大的已有事件，否则将其设为新事件(Allan et al., 1998)。该算法的关键步骤在于计算文本相似度，目前最为常用的是余弦相似度cosine。局部敏感哈希(LSH)聚类算法主要基于新事件检测模型(FSD)，其思路是通过LSH算法找到新来文本在已聚类文本中的近邻文本集合，从该集合中找到新来文本的最近邻文本，若两者最大相似度大于设定阈值，则为已有事件，否则为新事件(Petrović et al., 2010)。在局部敏感哈希(LSH)聚类算法中，其关键步骤在于通过LSH算法尽快找到新来文本的近邻文本。Wurzer et al. (2015)对寻找最近邻文本的哈希算法做了改进，提高了效率，但准确率与Petrović et al. (2010)相当。Xie et al. (2016)提出了一种基于自训练的深度嵌入式聚类模型。该模型使用深度神经网络同时学习特征表示和聚类分配，是一种基于划分的聚类模型，不适合处理流式数

据。Hadifar et al. (2019)使用SIF Embedding进行句子表征，采用自编码器提取文本的低维特征表示，采用类似Xie et al. (2016)的聚类算法通过自训练的神经网络模型同时学习文本特征表示和聚类分配，得到聚类结果。Finley and Joachims (2005)利用有监督的SVM模型判断两个文本是否相关，通过Bansal et al. (2004)的方法进行文本聚类，该聚类方法将样本分布视为一个图模型，通过最大化簇内的样本对相似性实现聚类。Haponchyk et al. (2018)将Finley and Joachims (2005)的方法进行改进，用于对话系统中用户问题的聚类，从而分析用户意图。Wang and Zhang (2017)采用了有监督的LSTM模型提取文本特征，计算文本相似度，采用增量聚类算法进行社交文本聚类，相比此前的聚类算法有所提高，但需要大量的数据对模型进行训练。目前，在面向社交媒体的事件聚类方法中，还没有采用半监督的方法。

3 半监督的社交媒体事件增量事件聚类模型（SemiEC）

有监督聚类算法虽然聚类效果较好，但需要大量的数据标注工作来训练模型提高效果。无监督方法的性能又往往不能满足实用的需求。基于此，本文提出了一种半监督的中文社交文本增量事件聚类模型（SemiEC），相比Wang and Zhang (2017)采用的模型在相同训练的情况下进一步提高聚类效果。

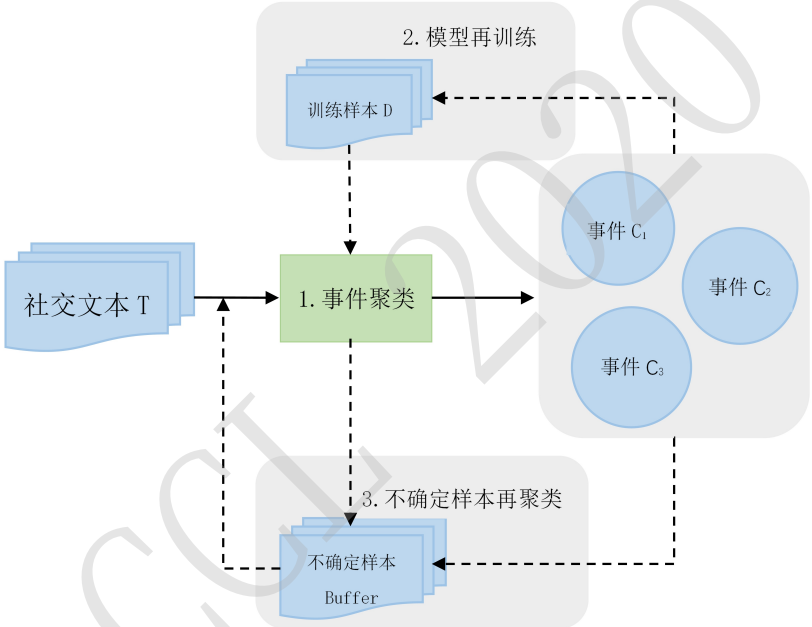

图 1: 基于数据增强的增量聚类过程

聚类过程如图1，对于输入的社交媒体文本t_i，首先对其进行事件聚类，判断t_i是属于已有事件还是新生事件，或是无法确定。若t_i属于已有事件，则将其加入该簇；若为新事件，则基于t_i建立新的簇，否则设t_i为不确定样本，加入Buffer。该过程中使用LSTM提取文本特征，利用线性模型计算文本相似度。由于增量聚类算法可以得到实时的聚类结果，因此每次对部分数据进行聚类后，从聚类结果中抽取样本组成训练集对模型进行再训练，使模型进一步学习新的事件特征，增强聚类效果。在结束后，将聚类结果中所含元素较少的簇中的样本全部设为不确定样本。然后用经过多次再训练后的模型对不确定样本进行聚类，得到最终的聚类结果。下面对聚类过程进行详细介绍。

3.1 文本表示

类似Wang and Zhang (2017)，SemiEC采用LSTM模型提取文本特征，记为$M_{encoder}$。在文本输入前，首先进行分词和去停用词。将分词后的文本用$X = \{w_1, w_2, \ldots, w_n\}$表示。其中$w_i$表示句子中第$i$个词在词表中的编号，$n$表示句子中的词数。使用由百度百科预训练的中文词向量对句子进行词嵌入，用n维词向量x_i表示词w_i，$x_i \in \mathbb{R}^{n*d}$。将文本向量$X = (x_1, x_2, \ldots, x_n)$通过LSTM模型得到一个隐藏序列$\{h_1, h_2, \ldots, h_n\}$，其中$h_t$由当前输入向量$x_t$和前一时刻的输

出h_{t-1}计算得到，$t \in (1,n)$，即$h_t = LSTM\left(x_t, h_{t-1}\right)$。训练过程中初始状态的参数均为随机生成。我们采用低维向量$H = h_n$来表示文本X。

3.2 文本相似度计算

对于两个文本X_i和X_j，首先通过$M_{encoder}$得到两个文本的特征向量H_i和H_j。将向量H_i和H_j拼接后，通过一个线性层得到向量H_c，最后通过一维线性层，由sigmoid函数得到X_i和X_j的相似度P_c，$P_c \in [0,1]$，将相似度计算模型记为M_{sim}，计算过程如下式。

$$H_c = \sigma\left(W_c^h\left(H_i \oplus H_j\right) + b_c^h\right) \tag{1}$$

$$P_c = sigmoid\left(W_c H_c + b_c\right) \tag{2}$$

其中，$\oplus$代表两个向量的拼接，W_c^h，b_c^h，W_c，b_c为模型参数。

3.3 聚类算法

SemiEC模型的聚类过程主要分为三个步骤：1）：对社交媒体文本的事件聚类；2）：对模型的再训练；3）：对不确定样本的重新聚类。

- 对社交文本的事件聚类(算法1-12行)

对于一个新到来的社交文本t_i，要判断其与已有事件的最大相似度。SemiEC模型用事件簇中的前N个文本来表示这个簇，作为簇心，若样本数不足N，则将簇中全部样本作为簇心。将t_i与代表簇心的各个文本的相似度平均值作为t_i与该簇的相似度。设定两个阈值L和H，其中$0 < L < 0.5 < H < 1$。若t_i与目前已有簇的最大相似度大于H，则将t_i加入相似度最大的簇；若t_i与目前已有簇的最大相似度小于L，则以t_i建立一个新的簇；若t_i与目前已有簇的最大相似度在L和H之间，则认为t_i分配不确定，将t_i加入缓冲区$Buffer$，暂时不进行聚类。

传统singlepass算法采用一个阈值判断文本t_i属于已有事件还是属于新生事件，相比较而言SemiEC模型加入了不确定样本这一分类，可以防止较差样本影响簇心表征和再训练的质量。

- 对模型的再训练(算法13-17行)

将特征提取模型$M_{encoder}$和相似度模型M_{sim}拼接到一起同时训练。设置一个更新阈值U，每当有U个样本完成聚类，就从U个样本中抽取D组训练数据组成训练集，对模型进行再训练。训练集中正例与负例的比为1:1。正例抽取方法为：从U个样本中随机选取一个样本p，再从U个样本中与p同簇的样本集中随机选取一个样本q，样本p与样本q组成一组正例，标签置为1。负例的抽取方法为：从U个样本中随机选取一个样本p，再从U个样本中与p不同簇的样本集中随机选取一个样本q，若U中没有与p不同簇的，则从目前所有已聚类且与p不同簇的样本中随机选取一个样本q，样本p与样本q组成一组负例，标签置为0。若目前所有已聚类的样本中都没有与p不同簇的，则本次不进行训练。

相比Wang and Zhang (2017)的方法，SemiEC增加了模型再训练的步骤，利用增量聚类过程中产生的标注数据对模型进行再训练，可以使模型学习新事件的特征，进一步提高模型的泛化能力。

- 不确定样本进行重新聚类(算法18-32行)

在增量聚类结束后，经常会出现个别样本数特别少的簇，这是增量聚类模型经常容易出现的问题。若聚类结束后，出现包含样本数少于N的事件簇，则删除这些簇，并将这些簇中的样本加入缓冲区$Buffer$，$Buffer$中包含聚类过程中的不确定样本。增量聚类结束后，再对$Buffer$中的样本进行重新聚类。计算不确定样本与已有事件的最大相似度，若样本与已有事件簇的最大相似度大于0.5，则将其加入该簇，否则以该样本建立新的簇。

相比Wang and Zhang (2017)的方法，SemiEC增加了不确定样本重新聚类的步骤，一方面，可以防止不确定样本加入簇心影响事件簇的表征以及进入训练集对模型进行错误的训练；另一方面，在聚类结束后，模型经过多次训练后效果有所增强，可以对这些不确定的样本进行更准确的聚类。

Algorithm 1 半监督社交媒体事件增量聚类算法

算法开始

Input:

社交文本$T = \{t_1, t_2, \ldots, t_n\}$
阈值L和H，表征簇的文本个数N
特征提取模型$M_{encoder}$，相似度模型M_{sim}
更新阈值U，每次训练集样本数D，缓冲区$Buffer$

Output:

事件聚类结果$C = \{C_1, C_2, \ldots, C_k\}$

Initialize: 将t_1初始化为第一个簇$C = \{C_1\}$；$Buffer = \varnothing$
for each $t_i \in \{t_2, t_3, \ldots, t_n\}$ **do**
  对于文本t_i，利用$M_{encoder}$得到其特征向量X_i
  利用M_{sim}计算X_i和C中的每个簇C_m的相似度Sim_m
  得到与t_i相似度最大的簇C_r，$Sim_r = Max\,(Sim_m)$
  if $Sim_r > H$ **then**
    将t_i加入C_r
  else if $Sim_r \geq L$ **then**
    将t_i加入$Buffer$
  else
    将t_i设为新的簇，加入C
  end if
  if 有U个样本加入C **then**
    抽取D组数据组成训练集，对模型M_{sim}和$M_{encoder}$进行训练
    通过$M_{encoder}$更新C中簇的表示
  end if
end for
for each $C_i \in C$ **do**
  if $|C_i| < N$ **then**
    将C_i中样本加入$Buffer$，删除C_i
  end if
end for
for each $t_j \in Buffer$ **do**
  对于文本t_j，利用$M_{encoder}$得到其特征向量X_j
  利用M_{sim}计算X_j和C中的每个簇C_m的相似度Sim_m
  得到与t_j相似度最大的簇C_r，$Sim_r = Max\,(Sim_m)$
  if $Sim_r > 0.5$ **then**
    将t_j加入C_r
  else
    将t_j设为新的簇，加入C
  end if
end for
输出$C = \{C_1, C_2, \ldots, C_k\}$

4 实验部分

4.1 实验数据

本次实验的数据来自微博。采用与Wang and Zhang (2017)相同的规则搜集了关于40次不同地震事件的微博，共计10828个。采用30次地震事件作为训练数据，从中随机选取样本组成训练集。其中同一地震事件中的两个文本为正例，标签为1；不同地震事件中的两个文本为负例，标签为0。训练集中正例和负例比为1:1，总共抽取了400000组样本训练模型。剩余的10次地震事件数据作为测试集，共包含2518个文本，用于事件聚类实验。

4.2 实验参数

聚类算法中N的取值为25，用事件簇中的前N个文本来表示该簇。N越大，对事件簇的代表性越强，但计算量也会增大。聚类过程中的阈值L和H的的取值范围为$0 < L < 0.5 < H < 1$，L的取值越接近0，H的取值越接近1，对不确定样本的筛选越严格，但与此同时也会增大计算量。本次实验中L的取值为0.3，H的取值为0.7。更新阈值U的取值不能太小，否则训练过于频繁，使模型容易受一些极端值的影响，产生偏离；U值过大则会使大量样本仅能使用原始模型聚类。本次实验中U值为200。每次训练集样本数D的取值越大，模型对事件特征的学习效果越明显，但同时也会增大运算量。同时D也受U的影响，实验中训练集样本数D取值为800，为U的4倍。训练轮数为5轮。

4.3 模型参数

模型基于Keras框架，后端为Tensorflow。特征提取模型$M_{encoder}$训练过程中使用了由百度百科预训练的词向量，向量维度为300，嵌入层设置为不可训练。LSTM层输出维度为128，dropout=0.1，recurrent_dropout=0.1，不返回序列，其余为默认参数。相似度模型M_{sim}中的全连接层输出维度为128，激活函数为relu，其中还包含两个Dropout层，Dropout=0.1。

神经网络N的输出为y_i，真实标签为$\bar{y_i}$，为0或1。优化器为adam，采用交叉熵损失函数binary_crossentropy，计算步骤如下：

$$loss = -\frac{1}{N}\sum_{i=1}^{N} \bar{y_i}\log(y_i) + (1-\bar{y_i})\log(1-y_i) \tag{3}$$

4.4 模型训练

将$M_{encoder}$和M_{sim}拼接到一起，两个文本通过$M_{encoder}$得到特征向量H_i和H_j，将其作为M_{sim}的输入，组成一个计算文本相似度的孪生神经网络N，最终得到两个文本的相似度。

将神经网络N在训练集上训练5轮，将训练得到的参数赋予$M_{encoder}$和M_{sim}，从而实现模型的预训练。模型再训练同样是对网络N进行再训练，训练轮数为5轮，将更新后的参数赋予$M_{encoder}$和M_{sim}，从而实现模型的再训练。

4.5 评价指标

我们采用纯度Purity，归一化互信息NMI和调整兰德系数ARI作为聚类效果的评价指标。Purity是正确计算的文本数与文本总数的比值。其定义如下：

$$Purity\left(\Omega, C\right) = \frac{1}{N}\sum_{k} \underset{j}{max}\left|w_k \cap c_j\right| \tag{4}$$

其中N表示总的样本个数，$\Omega = \{w_1, w_2, \ldots, w_K\}$表示聚类模型得到的聚类簇划分，$C = \{c_1, c_2, \ldots, c_J\}$ 表示真实类别划分。Purity取值范围为$[0, 1]$，越接近1，聚类效果越好。

NMI是一个基于熵的评价指标，其定义如下：

$$NMI\left(\Omega, C\right) = \frac{I\left(\Omega; C\right)}{\left[H\left(\Omega\right) + H\left(C\right)\right]/2} \tag{5}$$

其中I表示互信息:

$$I(\Omega;C)=\sum_k\sum_j P(w_k\cap c_j)\log\frac{P(w_k\cap c_j)}{P(w_k)P(c_j)}=\sum_k\sum_j\frac{|w_k\cap c_j|}{N}\log\frac{N|w_k\cap c_j|}{|w_k||c_j|} \tag{6}$$

H表示熵:

$$H(\Omega)=-\sum_i\frac{w_i}{N}\log\frac{w_i}{N} \tag{7}$$

NMI的取值范围为$[0,1]$，越接近1，聚类效果越好。
调整兰德系数ARI弥补了兰德系数RI惩罚力度不够的问题，其定义如下:

$$RI=\frac{a+b}{C_N^2} \tag{8}$$

$$ARI=\frac{RI-E[RI]}{max[RI]-E[RI]} \tag{9}$$

其中，a表示实际类簇与聚类预测类簇中都是同类别的元素对数，b表示实际类簇不同类别，在聚类预测类簇中也是不同类别的元素对数。$E[RI]$为RI的平均值。ARI的取值范围为$[-1,1]$，越接近1，聚类效果越好。

4.6 聚类结果

我们将提出的事件聚类模型得到的聚类结果与以下聚类方法进行对比。

1) *Singlepass*：该聚类方法采用向量空间模型表示文本，用cosine计算文本相似度，采用singlepass算法进行聚类，属于无监督聚类算法。模型由自己实现。

2) *Kmeans*：该聚类方法采用向量空间模型表示文本，用cosine计算文本相似度，采用Kmeans算法进行聚类，属于无监督聚类算法。实验中指定正确的事件个数。模型由自己实现。

3) *LSH*(Petrović et al., 2010)：这是一种基于局部敏感哈希的聚类算法，采用向量空间模型表示文本，用cosine计算文本相似度，属于无监督聚类算法。模型由自己实现。

4) *Hadifar*(Hadifar et al., 2019)：该聚类模型采用SIF Embedding表示文本，通过自编码器学习文本的低维特征表示，采用类似Xie et al. (2016)的深度聚类算法进行短文本聚类，属于无监督聚类算法。该方法在实验中采用不同领域的短文本作为测试集，本次实验中的测试集为地震领域的不同事件。实验中指定正确的事件个数。模型采用了改论文中提供的代码。

5) *Wang*(Wang and Zhang, 2017)：该聚类模型利用LSTM提取文本特征，利用线性神经网络模型计算文本相似度，通过增量聚类算法进行聚类，属于有监督聚类算法。模型由自己实现。

6) *BERT*：该方法将Wang and Zhang (2017)的模型换成了BERT词向量，属于有监督聚类算法。模型由自己实现。

由于聚类结果会受数据输入顺序的影响，因此我们将测试数据随机打乱顺序进行10次聚类，记录了各项聚类指标的平均值，聚类结果如表1。其中Wang and Zhang (2017)采用的有监督聚类模型与其他聚类模型相比聚类效果较好，因此以该模型为baseline。在经过相同训练的情况下，SemiEC模型相比baseline模型，在各项聚类指标上均有所提升，在Purity上提升3%，在NMI上提升4%，在ARI上提升10%。

社交媒体中的文本较短，表达具有随意性，即使对于同一事件的评论，其表述方式也各有不同。采用向量空间模型和词向量加权提取文本特征，都容易受干扰词的影响而导致关键特征信息无法突出，从而导致聚类效果较差。

通过对模型进行有监督的训练，相比无监督聚类模型，可以更准确的识别文本的事件特征，增强聚类效果。但基于BERT词向量模型的聚类方法相比基于word2vec词向量的方法聚类

聚类模型	Purity	NMI	ARI
Single-pass	0.54	0.56	0.39
Kmeans	0.76	0.75	0.65
LSH	0.48	0.37	0.20
Hadifar	0.50	0.45	0.34
Wang	0.78	0.79	0.60
BERT	0.63	0.60	0.47
SemiEC	**0.81**	**0.83**	**0.70**

表 1: 聚类结果对比

结果反而有所下降，原因在于BERT词向量模型考虑了更多的语义信息，而社交媒体中即使属于同一事件的文本语义描述也各有不同，因此反而导致结果较差，且BERT词向量模型相比word2vec词向量模型用时较多，对于海量的社交媒体文本而言效率偏低。

5 分析

5.1 训练数据大小对SemiEC模型的影响

有监督聚类算法对训练集的依赖较大，要使模型可以更准确地区分各种事件，其关键在于训练集中是否有足够充分的事件类型。因此，分别选取10次，15次，20次，25次，30次地震事件作为训练数据，从中抽取400000组文本对组成训练集对模型进行训练，以4.1节中的10次地震事件作为测试集，以Wang and Zhang (2017)的模型为baseline，以NMI为参考指标，将测试数据随机打乱进行10次聚类，对NMI取平均值，得到结果如图2。

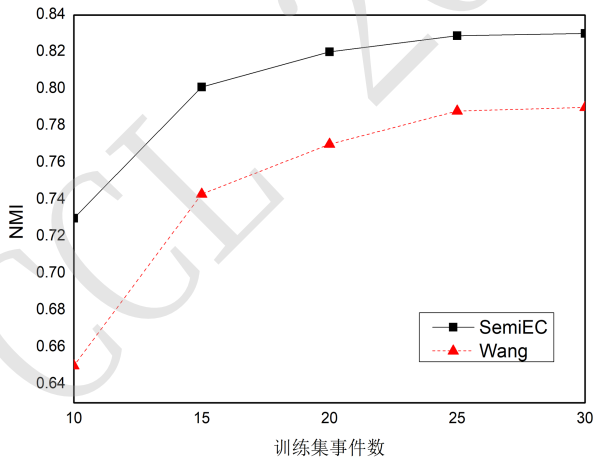

图 2: 不同训练集事件数对聚类结果的影响

由图2可以看出在不同训练集事件数的情况下，SemiEC模型相比baseline模型聚类效果有所提高，但不同训练数据的聚类效果有所差别。当训练事件数为15时，SemiEC的性能已经超过了使用更多训练数据的baseline模型。这充分说明了本文半监督方法的有效性，可以在少量标注数据的基础上，通过利用聚类过程中产生的标注数据学习新的事件特征，获得更好的性能。

5.2 参数设置对SemiEC模型的影响

模型再训练和不确定样本重聚类步骤都需要设置一些额外的参数。其中对SemiEC模型聚类效果影响最大的是更新阈值U，主要用于控制模型再训练的频率。本次实验将U分别设置为10，50，200，500，1000，每次训练集样本数D设置为U的4倍，分别为40，200，800，2000，4000。以4.1节中的10次地震事件作为测试集，以Wang and Zhang (2017)的模型为baseline，以NMI为参考指标，将测试数据随机打乱进行10次聚类，对NMI取平均值，得到结果如图3。

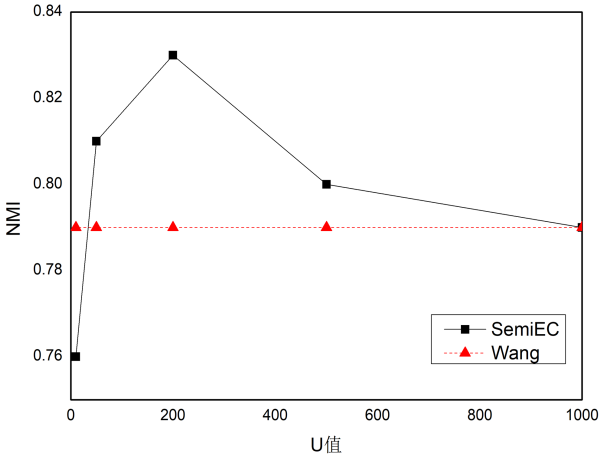

图 3: 不同U值对聚类结果的影响

由图3中结果可以看出，当U值较小，为10时，聚类效果较差，主要原因在于训练集较小，所属类别分布不均匀的概率较大，训练反而导致模型偏向于个别事件，使得聚类所得事件数比实际事件数少，聚类效果较差；当U值逐渐增大到200，样本分布趋于均匀的时候，聚类效果相对baseline会有明显提高；但不断增大U值会减少模型再训练的次数，导致大量数据仅能使用原始模型进行聚类，从而使聚类结果不断趋近但不低于baseline。因此，U的取值要在保证数据分布尽量均匀的情况下，取较小的值，此时可以使SemiEC模型达到最好的聚类效果。

5.3 模型再训练和不确定样本再聚类的有效性

为了证明模型再训练步骤和不确定样本再聚类步骤的有效性，分别对这两个步骤进行了测试。Retrain表示仅加入模型再训练步骤，Recluster表示仅加入不确定样本再聚类步骤。以4.1节中的10次地震事件作为测试集，以Wang and Zhang (2017)的模型为baseline，将测试数据随机打乱进行10次聚类，对各项聚类指标取平均值，得到结果如表2。

聚类模型	Purity	NMI	ARI
Wang	0.78	0.79	0.60
Retrain	0.79	0.82	0.69
Recluster	0.80	0.80	0.65
SemiEC	0.81	0.83	0.70

表 2: 模型再训练和不确定样本再聚类有效性对比

由表2数据可以看出，Retrain和Recluster，相比baseline在各项聚类指标上均有所提高，这充分说明了模型再训练和不确定样本再聚类步骤的有效性。其中，模型再训练步骤可以帮助模型学习新的事件特征，增强对后续样本的聚类效果。不确定样本再聚类步骤可以防止不确定样本加入簇心，减少错误样本对簇心表征的影响，从而增强聚类效果。将两者结合后的SemiEC模型，通过不确定样本再聚类减少错误样本进入训练集，增强模型再训练的效果，同时对不确定样本用再训练后的模型重新聚类，进一步增强不确定样本的聚类效果，两者相互提高，得到最好的聚类效果。

6 总结

本文提出了一种半监督增量型中文社交文本事件聚类模型SemiEC，采用LSTM提取文本特征，采用线性模型计算文本相似度，进行增量聚类。利用增量聚类过程产生的标注样本对模型进行再训练。对聚类过程中分配不确定的样本在结束后重新聚类。再训练过程可以让模型学习新的事件信息，使模型准确度随着聚类过程不断提高。对不确定样本的重新聚类可以防止不确定样本影响簇心表征，减少错误样本对模型进行再训练的概率，同时提高不确定样本的聚类准确度。SemiEC模型与经过同样预训练的有监督聚类模型相比，在各项聚类指标上均有所提高。

第十九届中国计算语言学大会论文集，第634页-第644页，海南，中国，2020年10月30日至11月1日。

参考文献

Charu C Aggarwal and Karthik Subbian. 2012. Event detection in social streams. In *proceedings of the SIAM International Conference on Data Mining*, pages 624–635. SIAM.

James Allan, Jaime G Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study final report.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2019. A simple but tough-to-beat baseline for sentence embeddings. In *proceedings of 5th International Conference on Learning Representations, ICLR 2017*.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. 2004. Correlation clustering. *Machine learning*, 56(1-3):89–113.

Deng Cai, Xiaofei He, and Jiawei Han. 2005. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637.

Thomas Finley and Thorsten Joachims. 2005. Supervised clustering with support vector machines. In *proceedings of the 22nd International Conference on Machine Learning*, pages 217–224.

Amir Hadifar, Lucas Sterckx, Thomas Demeester, and Chris Develder. 2019. A self-training approach for short text clustering. In *proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 194–199.

Iryna Haponchyk, Antonio Uva, Seunghak Yu, Olga Uryupina, and Alessandro Moschitti. 2018. Supervised clustering of questions into intents for dialog system applications. In *proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2310–2321.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Rui Li, Kin Hou Lei, Ravi Khadiwala, and Kevin Chen-Chuan Chang. 2012. Tedas: A Twitter-based event detection and analysis system. In *proceedings of 2012 IEEE 28th International Conference on Data Engineering*, pages 1273–1276. IEEE.

Michael Mathioudakis and Nick Koudas. 2010. Twittermonitor: trend detection over the Twitter stream. In *proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1155–1158.

Andrew J. Mcminn and Joemon M. Jose. 2015. Real-time entity-based event detection for Twitter. In *proceedings of International Conference of the Cross-language Evaluation Forum for European Languages*.

Duc T Nguyen and Jason J Jung. 2015. Real-time event detection on social data stream. *Mobile Networks and Applications*, 20(4):475–486.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to Twitter. In *proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189. Association for Computational Linguistics.

Cao Qimin, Guo Qiao, Wang Yongliang, and Wu Xianghua. 2015. Text clustering using VSM with feature clusters. *Neural Computing and Applications*, 26(4):995–1003.

Zafar Saeed, Rabeeh Ayaz Abbasi, Muhammad Imran Razzak, and Guandong Xu. 2019. Event detection in Twitter stream using weighted dynamic heartbeat graph approach [application notes]. *IEEE Computational Intelligence Magazine*, 14(3):29–38.

Zhongqing Wang and Yue Zhang. 2017. A neural model for joint event detection and summarization. In *proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4158–4164.

Dominik Wurzer, Victor Lavrenko, and Miles Osborne. 2015. Twitter-scale new event detection via k-term hashing. In *proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2584–2589.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *proceedings of International Conference on Machine Learning*, pages 478–487.

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 62–69.

Jiaming Xu, Bo Xu, Peng Wang, Suncong Zheng, Guanhua Tian, and Jun Zhao. 2017. Self-taught convolutional neural networks for short text clustering. *Neural Networks*, 88:22–31.

Pengpeng Zhou, Zhen Cao, Bin Wu, Chunzi Wu, and Shuqi Yu. 2018. EDM-JBW: A novel event detection model based on JS-ID'F order and Bikmeans with word embedding for news streams. *Journal of computational science*, 28:336–342.