Iterative Strict Density-based Clustering for News Stream

Kaijie Shi[1,2], Jiaxin Shi[1], Yu Zhou[4], Lei Hou[1,3], and Juanzi Li[1,3]

[1] Department of Computer Sci. & Tech., BNRist, Tsinghua University, China
skj19@mails.tsinghua.edu.cn, {houlei, lijuanzi}@tsinghua.edu.cn,
shijx12@163.com, bryanzhou008@ucla.edu,
[2] Tsinghua Shenzhen International Graduate School, Tsinghua University, China
[3] KIRC, Institute for Artificial Intelligence, Tsinghua University, China
[4] Department of Computer Science. University of California, Los Angeles, USA

Abstract. News Streams are booming with the prosperity of the Internet, leading to increased demand for an efficient and effective news clustering method. Since news reports vary greatly in different countries, languages and news-topics, clustering diverse news has proven to be a big challenge for all researchers. The results of current clustering methods expose their inability to detect fine-grained topics. They tend to detect topics on a coarse-grained scale, resulting in clustering different fine-grained topics together.
In this paper, we propose Iterative Strict Density-based Clustering(ISDC), a new approach for detecting fine-grained topics in an evolving news stream. The main idea of ISDC is to keep every cluster as a high-density cluster throughout the news stream by iteratively splitting growing clusters. We further apply multilingual-sentence-bert instead of word embedding as the news encoder to improve the news representation quality. We conduct comprehensive experiments on two datasets and demonstrate the superiority of our proposed method.

Keywords: Streaming Clustering, Iterative Density-based Clustering, Fine-grained Topic Detection

1 Introduction

Topic Detection and Tracking (TDT) [?] is an information processing technology designed to help people cope with the increasingly serious Internet information explosion problem. It aims to automatically identify new topics and keep track of known topics in the information flow of news media. As a key link of TDT, stream clustering aims to find news topics in evolving data streams in one pass using a limited amount of memory.

Density-based algorithms are an important group of stream clustering. By adopting the online-offline paradigm [?] with micro-clusters which are defined as high-density clusters, density-based algorithms consist of two main steps: Firstly all samples are assigned to different micro-clusters online, then these micro-clusters are merged into final clusters through offline density clustering. However, density-based algorithms are not satisfactory in terms of accuracy because of their loose restriction. In the online period, since the algorithm only compares samples with existing micro-clusters centers, news of the same cluster may be different. In addition, the offline clustering step only loosely limits the distance between different micro-cluster centers. As a result, the differences of samples within one micro-cluster becomes transitive, which further reduces cohesion in the final clustering result. Therefore, a strict restriction to all samples is indeed necessary for high-quality clustering in fine-grained topic detection.

To alleviate these problems of density-based algorithms, we propose Iterative Strict Density-based Clustering (ISDC). ISDC maintains a cluster set which only contains *topic clusters*. A *topic cluster* is a cluster where the distance between each sample is less than a certain value. When a new samples arrives, we try to insert it into the nearest cluster. After this insertion, if the corresponding cluster is no longer a topic cluster, ISDC will split the cluster iteratively using DBSCAN [**?**] until all sub-clusters become topic clusters. Furthermore, ISDC will gradually decrease the time weight of outdated clusters. Through sequential updates and iterative spliting, we group similar samples together and keep dissimilar samples far apart. Compared with other algorithms, ISDC can better distinguish different topics while maintaining computational equilibrium. Experimental results show that the *topic cluster* constraint improves clustering cohesion and stability.

In this task, we first use multilingual-sentence-bert [**?**] to encode text. It achieves promising performance in the task of topic detection.

The contributions of this work are summarized below:

– We propose the concept of *topic cluster* and a stream clustering algorithm to improve the accuracy of fine-grained topic clustering. In addition, our out time weighting mechanism for news can effectively distinguish news that occur in different times.
– Experimental results show our method achieving remarkable performance.

2 Related Work

Researchers have investigated a variety of methods for stream clustering [**?**], [**?**]. Many algorithms like CluStream [**?**], StreamKM++ [**?**] are partition-based algorithms. In these algorithms, the number of clusters have to be predefined. This is not suitable for news topic detection. Though the algorithm is simple, it only restricts the distance between each sample and the cluster center, resulting in dissimilar samples being gathered together.

Grid based algorithms use the grid data structure, which divides the whole space into a number of cells. Then, these cells are clustered to form the clustering result.

Most density-based clustering methods adopt the online-offline paradigm. The paradigm tracks up-to-date news in real time online and calculates the clustering result offline. Based on this paradigm, researchers proposed many methods including, DenStream [**?**] and D-Stream [**?**].

Another widely used type of clustering methods is hierarchical clustering, which generates clusters by iteratively combining the closest or most similar two clusters. It has a very popular successor, BIRCH [?], which performs better in terms of time efficiency. Another type of aggregative clustering is the SinglePass algorithm [?].

Providing an appropriate text representation for topic detection is a challenging problem. With the development of deep learning, Word2Vec [?] was proposed to learn high-quality distributed vector representations of word embedding. Sentence-Bert [?] uses siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. In our experiment, we compared different text representations in detail and the multilingual-sentence-bert representation performed the best.

3 Preliminaries

3.1 Task formulation

We shall first explain some theoretical notions by defining the concepts. We formulate the problem of detecting fine-grained topics after introducing these concepts to readers: given a news stream $G = \{s_1, s_2, s_3 ... s_i\}$, the goal is to aggregate news into different topics $C = \{c_1, c_2, c_3 ... c_k\}$ incrementally in real time. According to the occurrence of events, we put these outdated topics in the topic cluster queue C to $O = \{c_1, c_2, c_3 ... c_n\}$ to make C more efficient.

3.2 Distance definition

We introduce the definitions of distance between (1)sample and cluster (2)two different samples. We adopt multilingual-sentence-bert [?] as the encoder to generate the embeddings of news data. The model takes a piece of text sequence as input and outputs a fixed dimension vector. The embedding of the cluster center is defined as the average embedding value of all the samples in the cluster.

$$c_k = \frac{\sum_{n=1}^{N} s_n}{n} \tag{1}$$

s_n is the sample in c_k. Then the similarity between s_i and c_k

$$cs_{ki} = \frac{c_k \cdot s_i}{\|c_k\| * \|s_i\|}, \tag{2}$$

s_i is the document embedding arriving at time i, c_k is the kth cluster in the topic cluster queue. Similarly, the similarity between two samples is

$$ss_{ik} = \frac{s_i \cdot s_k}{\|s_i\| * \|s_k\|} \tag{3}$$

In addition,we use the time decay function to characterize the process in which similarity of the articles change with time difference.

$$\gamma = e^{(\frac{-1*(|t_i - t_j|)}{h})^p * \log 2}, \tag{4}$$

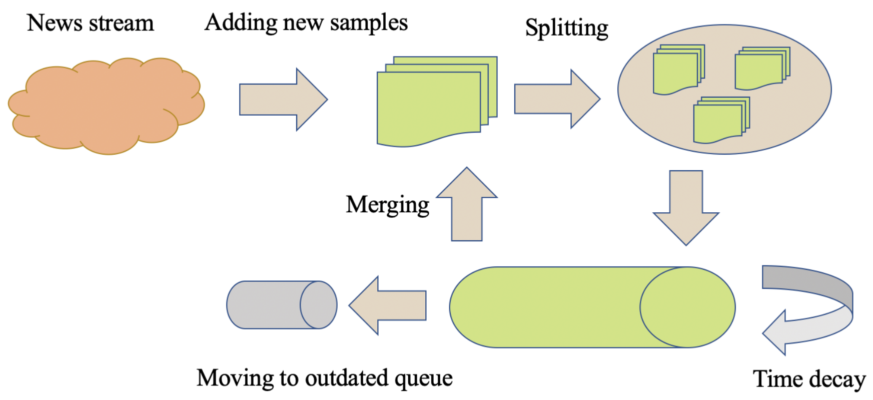


Fig. 1. Model Framework

where γ is the similairty of two times, t_i is the time document s_i arrives , t_j is the time when cluster c_j was created, h and p are two parameters that practically set by 15 and 1.8. In the experiment, we get the final distance between news s_i and cluster c_k

$$s_{ki} = 1 - \gamma * cs_{ki}, \tag{5}$$

s_{ki} is the distance to identify where we should insert the new sample.

4 Algorithm

In this section, we will go into detail and introduce our proposed Iterative Strict Density-based Clustering (ISDC). Figure 1 is our model framework. Our clustering algorithm consists of two parts: (1) Dynamic topic detection management (2) Outdated topic detection management.

4.1 Dynamic topic detection management

Adding new samples To discover clusters in an evolving news stream, we maintain a dynamic queue of *topic clusters*. When a new sample s arrives, the procedure of inserting it into the topic cluster queue is described below: 1. First we calculate the distances between the new sample and all existing clusters. If the distance to the nearest cluster is below our threshold δ, we insert the new sample into the nearest topic cluster c_p. Else we create a new topic cluster for this new sample.

Splitting and merging We set the cosine distance threshold as δ. As news event s_i arrives , we compare the distance between x_i and all existing topic clusters. After calculating those distances, we find the shortest cosine distance

Algorithm 1 Splitting and merging

```
Input:
$s_i$: next pending sample in the data flow
$\delta$:user's defined threshold
$\epsilon_0$: user's defined radius
$C$: topic cluster queue
DBSCAN(cluster, radius): density-based spatial clustering with noise [?]
Merge(sample, cluster): add the sample to the cluster.
Output:
$S$: the new topic cluster queue
 1: function "ITERATIVE CLUSTERING"
 2:     Select the nearest cluster $c_k$ from $C$, calculate the distance $d_i$
 3:     if $d_i <= \delta$ then
 4:         Add $s_i$ to the cluster $c_k$
 5:         if $c_k$ is not a topic cluster then
 6:             $\epsilon_0 \leftarrow \epsilon_0$ - 0.01
 7:             $sub \leftarrow$ DBSCAN(cluster=$c_k$,radius = $\epsilon_0$)
 8:             $iso \leftarrow c_0 \setminus sub$
 9:             for $s \in sub$ do
10:                 if $s$ is not a topic cluster then
11:                     goto Line 5
12:             for $p \in iso$ do Merge(sample=$p$,cluster=$sub$)
13:             add $sub$ to $S$
14:         return $S$
```

d_i between s_i and all topic clusters. If d_i is less than δ, we add x_k to the nearest cluster, else we initialize a new topic cluster with x_k. We restrict all clusters to be topic clusters, but the modified cluster is likely to violate the criteria due to inclusion of the new document. Hence, we check whether the modified cluster is still a topic cluster. If not, we shall split the cluster into several *topic clusters*.

In our method, we use DBSCAN [?] to split the modified cluster. DBSCAN defines clusters as the largest collection of densely connected points, it can divide regions by distance threshold. Let origin cosine distance threshold be the predefined ϵ_0. If the cluster isn't a topic cluster, in practice, we decrease ϵ_0 to ϵ_1 by 0.01. The aim is to divide the original cluster into more cohesive sub-clusters. If sub-clusters satisfy the criterion to be topic clusters, we stop splitting. If sub-clusters still fail to be topic clusters, then we have to iteratively split the sub-clusters until they become topic clusters. The number of iterations is constant because the lower limit of the cosine distance is 0. Since the number of iterations is constant, the time complexity of such iterations is O(n), an acceptable bound.

After splitting, the origin cluster is divided into several topic clusters. However, there are some isolated samples that may belong to the topic clusters. Hence we need to detect isolated samples and check if they belong to a topic cluster. We still use our previous methods of splitting: First calculate the distances between the isolated samples and topic clusters. If the distance is less than the original

Algorithm 2 Time Decay

```
Input:
$U$: the outdated queue
$decay$:decay function
$C$: the topic cluster queue
$p$: user's defined time period
$w$: user's defined weight threshold
1: function TIME DECAY
2:     for every time interval of $p$ do
3:         for $c_k$ in $C$ do time weight of $c_k$, $w_k = decay(w_k)$
4:             if $w_k < w$ then
5:                 move $c_k$ to $U$
```

threshold δ_0, we add the sample to the cluster, else we initialize a new cluster for the isolated sample.

4.2 Outdated topic detection management

Time decay For each existing topic cluster c_p, the weight will decay over time. The decay function is

$$w_p = w_p * 2^{-1*\sigma} \tag{6}$$

σ is the attenuation coefficient.

If w_p is less than w, it means that the cluster is outdated and should be moved to the outdated cluster queue. The outdated cluster queue is used to store all the history news events outside of the current time window. We periodically check and reduce the weight of the topic cluster. An important problem is how to determine the value of this time period. Generally speaking, news reports on a specific news event rarely last more than five days. Thus we set the period value as four or five days.

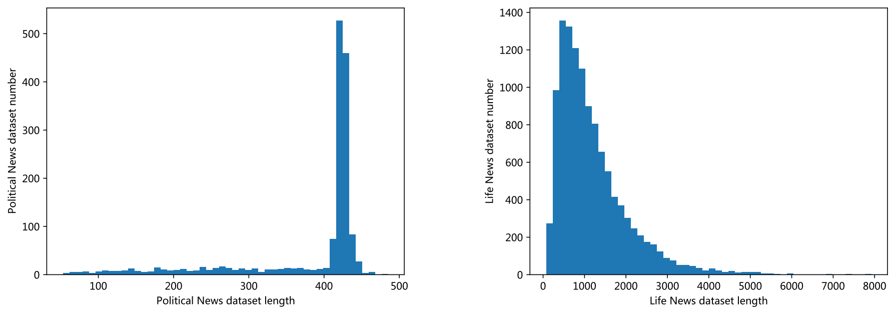
Moving to outdated queue Unike our algorithm, many other stream clustering algorithms adopt a different strategy. They adopt the online-offline strategy which store snapshots of the data stream and computes clustering results when necessary. Our method, on the contrary, combines the two steps into one. We directly compute the final clusters and adjust them dynamically. Then we merge the topic cluster queue and the outdated queue to get the final clustering result. This way, we reduce the computational pressure of the offline clustering process, and thus the calculation of the whole process is more balanced.

5 Experiment

5.1 Performance metrics

A good clustering algorithm requires all clusters to have high intra-cluster similarity and low inter-cluster similarity. Here are some common metrics in clustering. We use Purity, Silhouette Coefficient [?], FMI (Fowlkes–Mallows index) [?], and V-M. (V-measure) [?] as evaluation metrics for clustering results. Purity calculates the proportion of correctly clustered documents in total documents. FMI describes the difference between clustering result and the ground truth. V-measure is the harmonic mean of homogeneity and completeness, it comprehensively reflects the overall performance of the algorithm.

5.2 Dataset analysis



(a) Political News (b) Life News

Fig. 2. News Length Comparison.

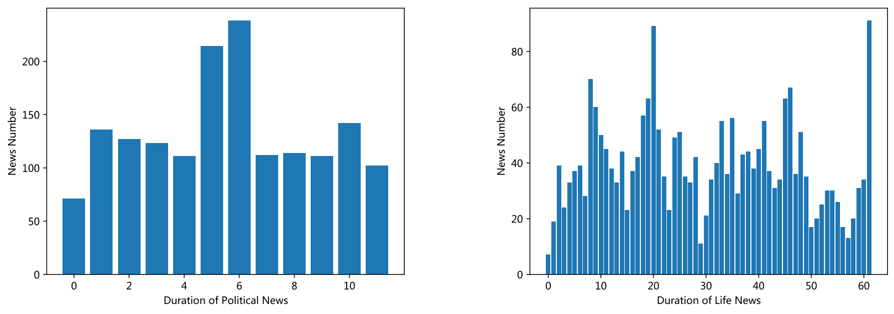
To ensure the comprehensiveness of our experiment, we experimented on two test datasets with completely different data distributions. We made a detailed comparsion of the two datasets in Figure 2 and Figure 3 The first news dataset is from a data mining system NewsMiner [?], an online news discovery and mining website. We selected all documents about political figures in the U.S. in 7 days, then our experts manually classified them into different categories.

The second test set is from *Growing story forest online from massive breaking news* [?]. The total number of news is 11748, with an average length of 1210.5 words. This news set contains news in many fields, including finance, sports, weather forecast, etc.

The length of each news piece in the Political News Dataset is concentrated in around 400-450 words, while lengths of news pieces in the Life news Dataset are more scattered, with lengths ranging from 0 to 5000 words.

5.3 Comparison with baseline algorithms

Parameter settings We experimented on the two datasets to compare different clustering methods. We compared our method with 4 clustering algorithms including BIRCH [?],SinglePass [?], DenStream [?], SOStream [?]. The common



(a) Political News (b) Life News

Fig. 3. News Time Duration Comparison.

parameters in the experiment are a) T , the cluster merge threshold used in density-based algorithms, b) N the predefined cluster number used in hierarchical clustering algorithms. We optimize these parameters separately using grid serach. Grid search not only ensures that comparisons between the unsupervised methods are fair, but also gets the best achieveable results of each individual method. Optimal results for each method are shown in the chart below.

Experimental results Our experimental strategy consists of two steps: First we run different clustering methods and obtain their respective aggregated samples. Then, we assign a new cluster to every isolated sample. For fairness of comparison, we use multilingual-sentence-bert [**?**] as the embedding model for all clustering algorithms.

Figure 4 displays a comparison of the clustering algorithms in terms of purity and Silhouette Coefficients. Purity is defined as

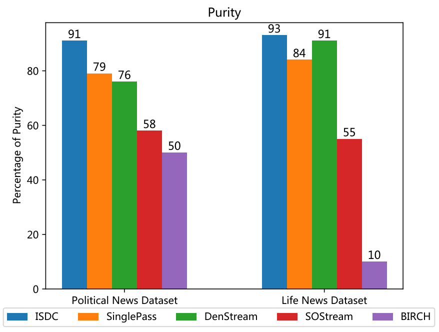
$$Purity(\Omega, C) = \frac{1}{N}\sum_{k}\max_{j}|\omega_k \cap c_j| \tag{7}$$

N is the total number of samples, $\Omega = \{w_1, w_2, ..w_K\}$ is the predicted cluster set. $C = \{c_1, c_2, ..c_J\}$ is the true cluster set. We can see from the formula: the higher the purity, the better cohesion within the predicted cluster. Our ISDC strictly controls the distance between samples, so our method outperformed its counterparts in terms of purity on these two datasets, as expected.

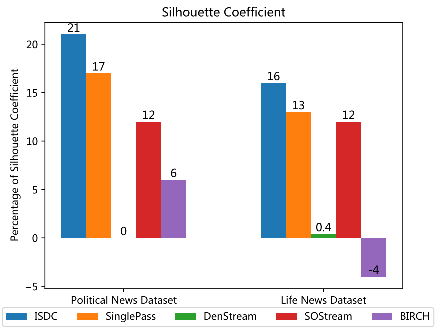
In order to exclude the impact of the annotated labels, we adopted an unsupervised indicator: the Silhouette Coefficient

$$SC = \frac{1}{N}\sum_{i=1}^{N} SC(d_i) \tag{8}$$

$$SC(d_i) = \frac{b-a}{max(a,b)} \tag{9}$$

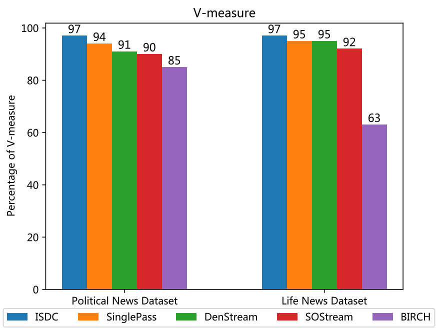


(a) Purity

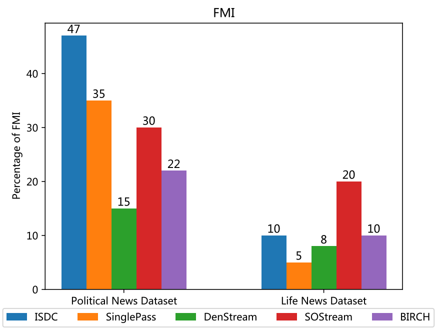


(b) Silhouette Coefficient

Fig. 4. Cluster Cohesion Comparison



(a) V-measure

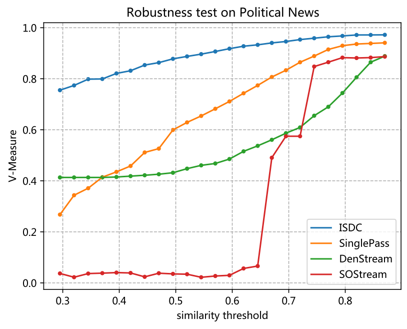


(b) FMI

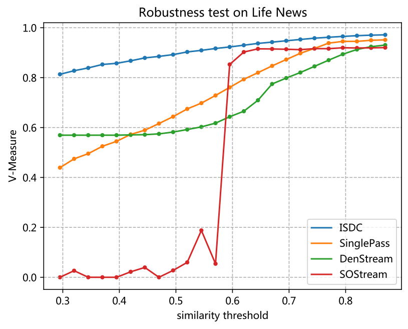
Fig. 5. Comprehensive performance comparison.

SC is the total Silhouette Coefficient, $SC(d_i)$ is the Silhouette Coefficient of cluster i. a is the average distance between a sample and other samples in its cluster, and b is the average distance between a sample and other cluster samples. The larger the Silhouette Coefficient is, the more compact the instances in the cluster are. Our clustering algorithm produces the most accurate reflection of difference in text embedding.

Figure 5 compares the algorithms with regard to two comprehensive indicators: V-Measure and FMI score. V-Measure is completely based on the conditional entropy between the two clusters, that is, after a certain category is divided, the uncertainty of the other category is determined. The smaller the uncertainty, the closer the two categories are divided. Therefore, the corresponding h value or c value is greater. V-measure is the harmonic mean of homogeneity and completeness, and it can more comprehensively reflect the effect of clus-



(a) Political News



(b) Life News

Fig. 6. Cluster Robust Comparison

tering. Due to our maintanence of the topic cluster queue, our method slightly outperforms others in terms of V-measure.

The Fowlkes-Mallows Index (FMI) [**?**] is defined as the geometric mean of the pairwise precision and recall:

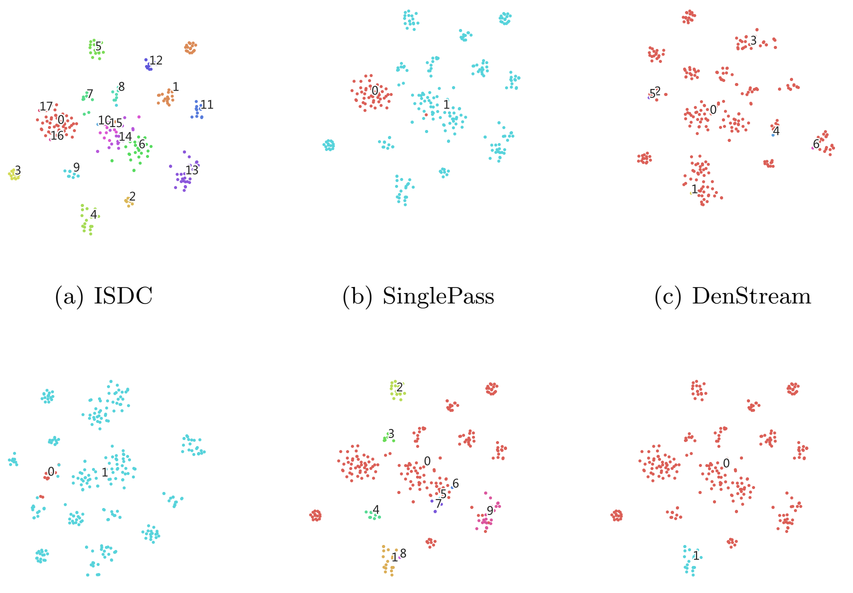
$$FMI = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}} \tag{10}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. Here the SOStream algorithms performs best among all algorithms. This is because FMI encourages samples to generate large clusters, and SOStream, with its loose clustering standards, performs well on this indicator.

In order to test the robustness of ISDC, we tested the performance of four algorithms under gradually changing parameters in figure 6. Different algorithms have different robustness. The SOStream's performance depends on the correctness of the parameter, and small similarity threshold will lead to a sharp decline in the performance of the algorithm. The SinglePass and DenStream algorithms have poor performance when the similarity threshold is small, but they have a rising performance when the similarity threshold is bigger. On contrast, our algorithm can stably achieve the optimal effect in different parameters, which can be seen as strong robustness.

5.4 Visualization of clustering algorithms

In order to visualize the differences between ISDC and other clustering algorithms, we randomly selected 20 categories from the test set, and then we used different algorithms to generate the cluster result with the same distance parameter. We used TSNE [**?**] to display their clustering results in Figure 7. Different colors represent different clusters generated by the algorithm. It's obvious that



(a) ISDC (b) SinglePass (c) DenStream

(d) SOStream (e) BIRCH (f) Agglomerative Cluster

Fig. 7. clustering results by TSNE

other algorithms can hardly recognize the differences between similar clusters while our method ISDC can successfully detect those clusters.

5.5 Ablation study on the embedding model

We tried three embedding models for our news topic detection task in Table 1. While using ISDC as our clustering algorithm, we tested Word2Vec, GloVe, M.S.(multilingual-sentence-bert) and compared their performances. The Word2Vec model and the GloVe model were pretrained on the news corpus in our Newsminer system.

Based on BERT, the multilingual-sentence-bert(M.S.) was fine-tuned with STS(Semantic Textual Similarity) and NLI(natural language inference). Table 1 shows that Word2Vec and GloVe have almost the same performance, while Sentence Transformer clearly outperforms the two. The M.S. produces even more exceptional results under the metrics of Purity, AMI and FMI.

Table 1. ISDC performace with different sentence embedding

Chinese News	Purity	AMI	FMI	Homo.	Comp.	V-M.
Word2Vec	0.83	0.32	0.20	**0.97**	0.95	0.96
GloVe	0.83	0.32	0.19	**0.97**	0.95	0.96
M.S.	**0.96**	**0.42**	**0.35**	**0.97**	**0.99**	**0.98**
Political News	**Purity**	**AMI**	**FMI**	**Homo.**	**Comp.**	**V-M.**
Word2Vec	0.86	0.43	0.28	0.95	0.96	0.95
GloVe	0.87	0.44	0.27	0.95	0.96	0.95
M.S.	**0.98**	**0.76**	**0.61**	**0.97**	**0.99**	**0.98**

6 Conclusion

In this paper , we describe the task of News Topic Detection(NTD). To accomplish this task, we describe a clustering algorithm which can generate *core clusters* by iteratively using the DBSCAN algorithm. In comparison to other baseline algorithms, our method achieved outstanding performance on the NED task while maintaining a simple structure. We also made a detailed discussion about the performance of ISDC and other algorithms. Our method is more robust and performs better. In the ablation study on the Embedding model, we found that the multilingual-sentence-bert [**?**] has a significant advantage over Word2Vec and GloVe. Although we have achieved promising experimental results, accuracy problems do occur when text representation of news is not accurate enough. We will further explore this issue in our future works.

7 Acknowledgements

This work is supported by the NSFC Key Project (U1736204) and grants from the Institute for Guo Qiang, Tsinghua University (2019GQB0003) and HUAWEI Inc.