A Clustering Approach Using Enhanced K-Means in 5G Networks

Min Zhu[1,2], Xin Xia[3], Jianming Zhang[4], Dengyin Zhang[1]

[1] School of Internet of things, Nanjing University of Posts and Telecommunications, China
[2] School of information technology, Zhejiang Shuren University, China
[3] School of Computer Information Technology, Wuhan Institute of Shipbuilding Technology, China
[4] School of Computer and Communication Engineering, Changsha University of Science and Technology, China
zhumin@zjsru.edu.cn, xinxinfavor@gmail.com, jmzhang@csust.edu.cn, zhangdy@njupt.edu.cn

Abstract

There have been amount of messages for multiple users that exist in 5G communication networks. In order to provide high-quality services for these users at the same time, the messages for different users need to be grouped by clustering. The clustering approach has been promoting the growth of smart businesses in transmitting applications. For a large-scale data, applying this mining algorithm is severely limited. In this paper, we focused on the K-Means clustering where the Frequent Pattern-growth (FP-growth) algorithm applied to each cluster. The mined frequent items of the short text always have the similar meaning. And the representative candidates based on frequent items can represent the whole cluster of short texts that they belong to. This method of an enhanced K-Means based on FP-Growth breakthrough the limit of assuming the number of clusters, k, known in advance. It can automatically find as many candidates of short sentence as possible, and runs much faster and consumes much less main memory than the general algorithm by the computing distances between text documents like Earth Mover's Distance based on Word2Vector.

Keywords: Clustering, Short text clustering, K-Means, FP-Growth

1 Introduction

With the development of 5G technology, more and more users use 5G networks for communication. However, an inevitable problem is that when multiple users use the same frequency band to communicate at the same time, the communication quality will decline accordingly. Although many new technologies have been proposed, such as the application of massive MIMO, energy transfer, and NOMA technologies [1-4], alleviating the dilemma of insufficient spectrum, but from the users themselves, grouping multiple users through clustering services can effectively improve the spectrum.

Today amount type of messages for multiple users existing on the transmitting networks can be characterized with short text, e.g., social media transmitter, products reviewer, and instant messages [5]. In this paper, we focus on web application about job searching and auto resume screening. The text involved here is with the limited document length, typically only tens of words or even less. While clustering has been becoming one of the most common data mining method [6-7], the traditional clustering like K-means applied to short text sets still causes the problems [8]. First, it's how we set a suitable number of clusters. Some conventional clustering algorithms need to provision a parameter as the number of clusters in advance, such as K-means and Iterative Selforganizing Data Analysis Techniques Algorithm (ISODATA). Setting the parameters requires a deep understanding of data and searching an ideal number of clusters with heuristic methods often takes too much time. Second, the vector representing short text where most words only occur once usually is sparse and high-dimensional, which may lead to the performance of clustering is not good. Third, clustering relies on the computation of the similarity between each pair of short text snippets that belong to the same cluster. The similarity can be measured by the co-occurrence of terms between two short text snippets [9]. Unfortunately, this type of similarity metric is not suitable for short text snippets because of their sparsity, and it would not enhance the performance of clustering [10]. Finally, exploiting the semantic information of text, Kusner et al. calculated the distance of short text based on the deep learning model [11] Word2Vector and Earth Mover's Distance [12]. However, the computational complexity of is so high that it's unsuitable to big data.

The text data in this paper came from the recruitment websites about the required skills and

*Corresponding Author: Dengyin Zhang; E-mail: zhangdy@njupt.edu.cn
DOI: 10.3966/160792642020122107003

experiences by HR or employers. To select representative skills and experiences, the support of frequent itemsets and the important weight of TF-IDF are both designated. The skill description corresponding to frequent items below the threshold is not universal. We consider each reserved frequent item as the center of a sub-cluster. A sub-cluster consists of short documents with same frequent item-set. Two short texts clustering rules following the algorithm: Candidate Generation Rule and Merge Sub-Clustering Rule. The goal of Candidate Generation Rule is to obtain the candidate short text that best represents the sub-cluster. The Merge Sub-Clustering Rule merges sub-clusters that have the same candidate. These two rules finally determine the number of cluster and candidate of each cluster.

The main contributions of this paper are summarized as follows: (1) Proposed an enhanced K-Means algorithm based on FP-Growth for short text clustering. The algorithm can better find the suitable number of clusters and candidates as centroids of each cluster. (2) Designed Candidate Generation Rule to determine who is the candidate in each cluster. (3) Designed Merge Sub-Clustering Rule to merge sub-clusters by the same candidate.

2 Related Works

The application of clustering methods in wireless communication systems is still rare. However, the traditional clustering methods have been deeply studied. We first introduce the traditional clustering method, and then explain how we apply the improved clustering method to 5G networks. The typical mining technologies of document topics mainly cover the following aspects: finding similar items, frequent itemsets and clustering. Basically, we first need to find similar documents. In order to turn the problem of textual similarity of documents into one of set intersection, Manber used a technique called "shingling" [13]. A k-shingle (or k-gram) for a document is a sequence of k tokens that appears in the documents. Instead of TF-IDF, Shingling accounts for ordering of words.

Zhou et al. [14] proposed a method known as Frequent Itemsets based Clustering with Window (FICW). The experimental results showed that FICW had better performance for both clustering accuracy and efficiency [15]. Xiangwei Liu and Pilian Liu [16] proposed a new clustering algorithm, which is called Frequent Term Set-based Clustering (FTSC). It aplies frequent itemsets for clustering. Firstly, it detects important information from documents and sets it into databases. Then, it employs the Apriori to mine the frequent itemsets. Finally, it clusters the documents as per the frequent words in subsets of the frequent item-sets. The algorithm can reduce the dimension of the text data for extremely large databases, thereby it could improve the accuracy and speed of the clustering algorithm. The experimental results showed the superiority of FTSC and FTSHC algorithms over K-Means algorithm in the clustering performance.

To deal with the current Web document clustering, Wang et al. presented a new algorithm based on K-Means and top-k frequent term sets, which is named as simple hybrid algorithm (SHDC) in [17]. In particular, top-k frequent term sets provide K initial means and consider initial clusters and later refined by K-Means [18]. Then, K-Means return the final optimal cluster and K frequent item-sets gives the clear description of cluster. Experimental results showed that SHDC had better performance over the farthest first K-Means and random initial K-Means for both efficiency and effectiveness. A web-text clustering method is proposed by Su et al. in [19], which is based on maximal frequent item-sets for personalized e-learning. Firstly, it used vector space model to represent the web documents. Then, it determine the maximal frequent word sets. Finally, maximal item-sets were used for clustering documents, which is based on a new similarity measure of item-sets. Experimental results demonstrated that the presented method was excellent.

A frequent term based parallel clustering algorithm was introduced by Wang et al. in [20], which could be used to cluster short documents and large text datasets. An effective method to improve the accuracy of clustering is the semantic classification method. Experimental analysis proves that this algorithm has good performance for large short documents. In addition, the algorithm has good portability and extensibility, and can also be used to process documents with a large amount of data. Liu et al. Proposed a clustering algorithm based on frequent itemsets [21]. First, the documents were represented by a vector space model (VSM), and each term was ranked according to its relative frequency. Then use frequent pattern growth (FP-Growth) to mine frequent itemsets, and finally, cluster the documents based on the results of frequent itemsets. The experimental results show that this method is very effective for large databases, and the identified clusters are clearly explained based on the scientific research of frequently occurring itemsets.

Different texts have different subject and term pairs. These are difficult to obtain directly in traditional clustering algorithms. H et al. Proposed a new clustering algorithm based on the possibility of term pairs [22]. Get a term pair from a term set that corresponds to a related topic. Accordingly, the subject and its description also come from the term set. The collection of these term pairs is consistent with the meaning of the subject. Based on three benchmark text sets, the experimental results in [23] show that the method is efficient and has good performance.

It is worth noting that the similarity in the communication system we are concerned about is the

similarity of characters, not similar meanings. So we need to check the word and not the grammar of each document. In [12], Kusner et al. proposed a new concept of Word Mover's Distance (WMD). By defining the distance function between different documents, the expression form of learning words from local sentences is realized. WMD needs to measure the difference between two texts and the minimum distance between two embedded words. So their metrics can be implemented directly [24]. In addition, the experimental results show that the application of the WMD standard can significantly reduce the error rate of document classification.

Wang et al propose an offline feature extraction model [25], which takes the log event as input of word2vec to extract the relevance between log events and vectorize log events directly.

3 The Proposed Method

In this section, we will explain in detail the algorithm we proposed in the following five parts. In Section 3.1, we introduce the TF-IDF model to encode the document and use Euclidean distance to measure the similarity of different texts. The encoded feature vectors are clustered by using K-Means. In addition, we also give a method for labeling clusters in Section 3.2. In Section 3.3, the FP-Growth is used to efficiently search for frequent itemsets. In the process, we filter out some frequent itemsets with a threshold and select the most representative frequent itemsets. The remaining frequent itemsets are considered as another cluster center of sub-clusters. In other words, this sub-cluster consists of documents that containing this one remaining frequent itemset. Section 3.4 and Section 3.5 respectively introduce Candidate Generation Rule and Merge Sub-cluster Rule. Candidate Generation Rule is applied to determine optimal candidate of each sub-cluster and Merge Sub-cluster Rule is used to merge sub-clusters with same candidate. In addition, we will also discuss the details not mentioned in the above process, such as "one_term", a document that contain only one word. The algorithm flowchart for short text clustering is shown in Figure 1.

3.1 Text Model

We first introduce the general text model. m denotes the number of document. n denotes the number of terms representing non-repeating words appearing in the entire datasets. For convenience, we use ti and doc i to represent one of the terms and documents, and use the vector space model to represent documents, and each document is expressed as a vector. In this paper, we use the TF-IDF term weighting model, in which each document vector V is represented by

$$V = \mathrm{tf}(t_1)\times\mathrm{idf}(t_2),\ldots,\mathrm{tf}(t_n)\times\mathrm{idf}(t_n) \quad (1)$$

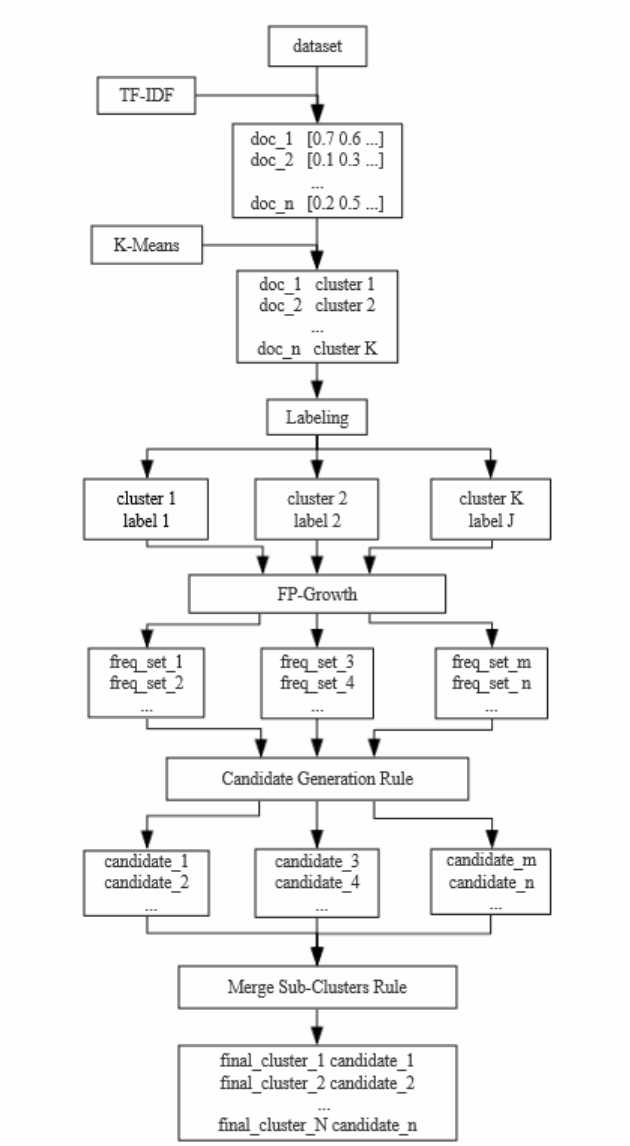


Figure 1. The algorithm flowchart of enhanced K-Means based on FP-Growth

where $\mathrm{df}(t_i)$ is calculated by the number of documents that contains the *i*-th term in the datasets. Consequently, the vector of a piece of document can be represented by

$$V = (w_1, w_2, \ldots, w_i, \ldots, w_n) \quad (2)$$

where w_i is the weight of the *i*-th term in the document. It can be expressed as follows. If t_i belongs to document, w_i equals $\mathrm{tf}(t_i)\times\mathrm{idf}(t_i)$, otherwise, set 0 to w_i.

We used Euclidean distance to measure the similarity between any pair of documents and the distance between each document and cluster center, where the cluster center is a vector with the same dimension as the document vector *V*. The Euclidean distance between doc *p* and doc *q* is defined as follows

$$\mathrm{distance}(p, q) = (w_{p1}-w_{q1})^2+\ldots+(w_{pn}-w_{qn})^2 \quad (3)$$

Where w_{pi} and w_{qi} are $\mathrm{idf}(t_i)$ in *p* and *q*, *n* is the number of terms, which equals to the dimension of the short text vector.

3.2 Short Text Clustering and Cluster Labeling

Due to the data preprocessing such as unified lowercase for all the characters in documents, some documents in the dataset may have more than one duplicate. The number of the copy of some document sometimes may be so large. It means the skill text expressed by that document is what a number of companies need and is more important than the small number of documents. But those copies are unfitted for the clustering here. When the copy number of a document is very high like occupying a large proportion, the cluster center would be decided by that document and move very close to it, which lead to neglect all other documents. So we count all repeated document before making them unique and use num_doc to denote the number of a repeated document, which will be used in Candidate Generation Rule (Section 3.4). Our algorithm uses K-means twice. At the first use, we are interested in the meaning that each cluster really stands for, such as technical skills, soft skills, experience, etc. Thus, we defined 9 labels to label each cluster, which is shown in Table 1. Each label is related to a set of words handpicked in advance. Then we examine if the set of words matches the meaning of each cluster by taking the ten largest values of word from the cluster centroid. For example, "responsibility", "communication" and "collaboration" belong to the set of the skills label. Since the value of wi reveals the importance of its corresponding words, each clustering is labeled by matching the words in label vocabulary with keywords, where the keywords are the words corresponding to the top ten wi of the cluster centroid. If more than half of the keywords in one cluster appear in a label vocabulary, we will use the label to mark this cluster. We merge the clusters with same label and drop the clusters with "noisy" label.

Table 1. Clustering' labels

Label No.	Label
1	Soft skills
2	Technical skills
3	Salary
4	Subject description
5	Welfare
6	Experience (years)
7	Education
8	Major
9	Noise

3.3 Frequent Itemsets Mining

Before talking about frequent itemsets, we consider one term, only a word in document. The data preprocessing not only generates the repeated documents as mentioned in Section 3.2, but also results in some documents that contain only one word. For example, when a document contains lots of stop words, removing stop words may lead to the document contain one word. The word of repeated documents may be very important skills if its amount is large enough. One word obviously is not suitable for frequent itemsets mining and we name that document of "one_term" and record its numbers and which cluster it belongs to.

Next we use FP-Growth algorithm to search frequent itemsets. Considering short texts, we assume the length of frequent itemset is rarely greater than 3. Although there is a frequent itemset with more than 3 elements. As we all know, if a superset contains all elements is frequent, then its subset is also frequent. Therefore, we set the length of the frequent itemset to 2. Setting the minimum support parameter r is to filter out the frequent itemsets whose number is too small. Furthermore, we also except the frequent itemsets that are not redundant, which means different frequent itemsets do not contain the same elements. For instance, frequent itemset p consists of word a and b, frequent itemset q consists of word a and c. We will remove a frequent itemset by comparing w_i of word b and word c. If the w_i of word b is large than word c, we dropout q, otherwise, p is removed. In fact, the candidate using Candidate Generation Rule (introduced in Section 3.4) may be the document that contains a, b and c, so there is no need to worry losing important skills.

Algorithm 1. Sub-cluster generation

Input: clusters, minimum support r, num_doc
Output: num_frequent_item-set, one_term sub-cluster of each cluster

Process:
1. Init one_term = {oneterm:[number, cluster]}
2. For cluster 1 to cluster K
3. For doc in cluster i
4. If doc i is oneterm and not in one_term
5. oneterm = doc i, number=1, cluster = cluster i
6. If doc i is oneterm and in one_term
7. number+=1
8. Using FP-Growth on each cluster with r
9. Sort frequent item-sets by descending order of w_i
10. # since a frequent item-set is center of a sub-cluster
11. For each doc in this cluster
12. If doc p contains the frequent item-set
13. num_frequent_item-set += num_doc p
14. sub-cluster append doc p
15. If doc p not contain any frequent item-set
16. Calculate distance from p to all frequent item-sets
17. Add p to sub-cluster with shortest distance

The remained frequent itemsets are regarded as the center of sub-clusters, where a sub-cluster is composed of documents containing some frequent itemset. Obviously, the number of frequent itemsets is the same as the number of sub-clusters. There may be a small

proportion of documents that do not have any frequent itemset. As for each of the document, we use expression (3) to calculate the distance from document to all frequent itemsets and add it to sub-cluster with minimum distance, that is, add the document to the nearest sub-cluster. In this way, we can ensure that all documents in a cluster are attributed to different sub-clusters except "one_term". In addition, we count the number of each frequent itemset and use num_frequent_item-set to represent it. Here, the number of a frequent item-set is to sum the number of whole documents containing the frequent itemset, where the number of documents, num_doc is got in Section 3.2. Since a document that do not contain any frequent itemset is very rare, num_frequent_itemset is almost equal to the number of documents in a sub-cluster. Therefore, we use num_frequent_itemset to represent the number of documents in a sub-cluster, which will also be used in Candidate Generation Rule. Algorithm 1 is the pseudo-code description of the process above.

3.4 Candidate Generation Rule

We could use Algorithm 1 to obtain all frequent itemsets and the corresponding sub-cluster of the entire document dataset. However, "one_term" still has not be processed. We want to know if the skill candidates containing "one_term" are already included in a sub-cluster. In addition, there are some sets that cover more than one frequent itemset. So, the sub-clusters corresponding to these frequent itemsets can be merged into a larger cluster. We finally expect to determine a representative document that represents the sub-cluster as a candidate rather than a frequent itemset. The above process is depicted by Candidate Generation Rule (the pseudo-code listed in Algorithm 2).

Algorithm 2. Candidate Generation Rule

Input: sub-cluster, scale factor *s*, minimum quantity threshold *x*, num_doc, num_frequent_item-set

Output: candidate of each sub-cluster

Process:

1. For each sub-cluster
2. Select doc with max num_doc
3. If max num_doc>*s*×num_frequent_item-set
4. candidate = the doc with max number
5. Else
6. Find all doc whose number bigger than *x*
7. Calculate distance from doc to center
8. # center is the frequent item-set of this sub-cluster
9. candidate = doc with min distance

Generally speaking, the candidate is either the document with largest number or the document closest to the center. So, in Candidate Generation Rule, we use two conditions to select candidates. For Condition 1, a document chosen as a candidate must be large enough such that it can account for a certain proportion of the entire sub-cluster. The scale factor s is to measure whether the document meets requirement. On the other hand, we need to pick the document with the biggest number from each sub-cluster. When the number is bigger than num_frequent_itemset times s, we consider that document as a candidate. Otherwise, we use condition two to find candidate. For Condition 2, we create a minimum threshold x. All documents larger than x have an opportunity to be a candidate. Then, from them, the document nearest to the center will be a selected candidate, where the distance is calculated by expression (3).

3.5 Merge Sub-Clustering Rule

In this part, we mainly deal with two situations mentioned in Section 3.5. First, for the relationship between "one_term" and sub-cluster, "one_term" can be used to match the words in candidates of a cluster, as we have known each "one_term" belongs to which cluster. Once one_term appears in a candidate, it is merged into the sub-cluster corresponding to the candidate. It will be treated as an independent cluster if its amount is large enough such that it exceeds the threshold. Second, different frequent itemsets may have the same candidate. We just merge these frequent itemsets and treat the candidate as a big frequent itemset whose elements is union of the elements in those frequent itemsets. Finally, the final candidates and final clusters are obtained. The pseudo-code of Merge Sub-Clustering Rule is given in Algorithm 3.

Algorithm 3. Merge Sub-Clustering Rule

Input: candidate of each sub-cluster, sub-cluster of each cluster, one_term, one_term output threshold *y*

Output: final candidate and final clusters

Process:

1. For each cluster
2. If there is only one sub-cluster
3. candidate = candidate of the sub-cluster
4. For one_term in this cluster
5. If different sub-cluster have the same candidate
6. merge_candidate = candidate shared by sub-cluster
7. initial_candidate = candidate not shared by sub-cluster
8. For one_term in this cluster
9. If one_term not in merge_candidate and not in / initial_candidate and its number > *y*
10. one_term_candidate append one_term
11. cluster_candidate = merge_candidate U / initial_candidate U one_term_candidate
12. final_candidate = merge all cluster_candidate
13. a final cluster is all document with a final candidate

4 Experimental Results

4.1 Datesets

The dataset crawled from the job-seeking websites (www.zhaopin.com, www.seek.com.au) includes over 20 million items. From them, we selected about 120,000 items relating to job employment including Java engineer, Software tester and Web developer. Each item contains the attribute of job description evolving the text of job skills. Our goal is to extract the technical and soft skills from large-scale job description set. First, establish job requirements thesaurus using keywords search, and extract the job requirements from the job description text. Second, extract the shorter sentences of long-formatted job requirements. Third, use the word segment tool to split the shorter sentences into separate words. Finally, further remove stop word and unify all letters to lowercase for each shorter sentence in order to generate a standard data format. Figure 2 is a piece of job recruitment one the Web. Table 2 shows a list as part of job positions we provisioned, the data between positions could overlap each other.

Requirements

- 3-5 years of hands-on experience with Java/J2EE and object-oriented programming
- Knowledge of software development principles and design patterns
- Experience with web application standards and technologies – Java, J2EE, JSF, JDBC, XML, XSLT, JAXB, Spring, Hibernate, etc.
- Exposure to one or more of the following application servers – Tomcat, JBoss, WebLogic, WebSphere
- Familiarity with SQL and relational (MS-SQL, Oracle, Postgres, etc.) or NoSQL (HBase, Cassandra, MongoDB, etc.) databases
- Exposure with one or more of the following build and deployment tools – Ant, Maven, Jenkins, Gradle, Ivy, Puppet or Chef
- Exposure with source control tools (e.g. Subversion, GIT)
- Bachelor's Degree in Computer Science, Computer Engineering or a related technical filed from a top school
- Travel may be required

Figure 2. Skill description of job recruitment from job-seeking websites (www.zhaopin.com,www.seek.com)

Table 2. List part of job positions for experiments

Position name	Number (thousand)
junior Java developer	33,589
Mid-Level java developer	40,344
senior java developer	29,230
software tester	11,782
web developer	23,773
...	...

4.2 Evaluation

In the experiment, the parameter K, the number of cluster of K-Means initially set to 9. The value of the minimum support r in the FP-Growth algorithm set to 5(Algorithm 1). In Candidate Generation Rule, s set to 0.25 and x set to 10. The "one_term" output threshold y set to 100 in Merge Sub-Clustering Rule. Next we further set the value of parameter K from 10 to 17 and finally obtained the optimal number of clusters. Intuitively, no matter what value of K to be chosen, the final number of clusters hardly change. The experimental results are shown in Table 3.

Table 3. The number of "final" clusters with different initial K of K-Means

K	number of clusters
9	17
10	17
11	19
12	18
13	17
14	19
15	18
16	18
17	17

We referred to compactness (CP) as an indicator to express the clustering characteristic with different final number of clusters in Figure 3, where the compactness was calculated by the following steps. We defined this final average distance as compactness. If a cluster is the type of "one_term" that is not merged, the average distance of the clusters is 0. Because all the documents in these cluster are the same. The final number of clusters given by our algorithm, which is nearly 18, is located at the "elbow" point in Figure 3.

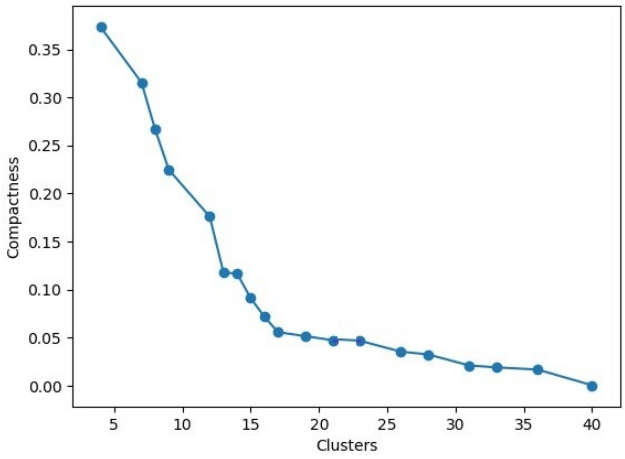


Figure 3. The relation between the number of cluster and compactness

Our algorithm takes 4 minutes at most to process over 70,000 pieces of data, while Word2Vector and Earth Mover's Distance model spend about 40 minutes. The comparison of time consumption is shown in Figure 4. Our algorithm could run efficiently, even if the amount of data is very large.

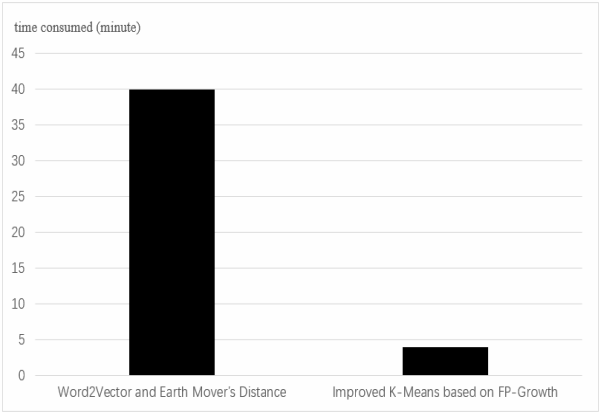


Figure 4. Comparison of two methods in time consumption

5 Conclusions

In this paper we proposed a new data mining algorithm for short text which uses the enhanced K-means based on FP-Growth. The motivation of this method is to cluster short text set with frequent itemsets which emphasizes the association strength between words instead of the similarity between short text like LSH [13]. Candidate Generation Rule determines which short text is the best candidate for each sub-cluster. Merge Sub-Clustering Rule merges the sub-clusters by the same candidate. Combination of our proposed algorithm with the two rules could find a better number of clusters. In addition, the algorithm still preserves efficiency when processing big data. Our algorithm is evaluated on the data about job positions and skills from job-seeking websites. The results show that we can obtain a lower inter-class distance than the traditional short text clustering algorithms.

Acknowledgments

This research was funded by the Young doctor innovation program of Zhejiang Shuren University under Grant Numbers: 2019QC30.

References

[1] J. Wang, Y. Gao, C. Zhou, R. S. Sherratt, L. Wang, Optimal Coverage Multi-Path Scheduling Scheme with Multiple Mobile Sinks for WSNs, *Computers, Materials & Continua*, Vol. 62, No. 2, pp. 695-711, 2020. DOI: 10.32604/cmc.2020.08674.

[2] S. He, Y. Tang, Z. Li, F. Li, K. Xie, H.-j. Kim, and G.-j. Kim, Interference-Aware Routing for Difficult Wireless Sensor Network Environment with SWIPT, *Sensors,* Vol. 19, No. 18, p. 3978, September, 2019. DOI: 10.3390/s19183978.

[3] J. Wang, W. Wu, Z. Liao, R. S. Sherratt, G.-j. Kim, O. Alfarraj, A. Alzubi, A. Tolba, A Probability Preferred Priori Offloading Mechanism in Mobile Edge Computing, *IEEE Access*, Vol. 8, pp. 39758-39767, February, 2020.

[4] J. Wang, W. Wu, Z. Liao, A. K. Sangaiah, R. S. Sherratt, An Energy-Efficient Off-Loading Scheme for Low Latency in Collaborative Edge Computing, *IEEE Access*, Vol. 7, pp. 149182-149190, October, 2019.

[5] J. Wang, Y. Yang, T. Wang, R. S. Sherratt, J. Zhang, Big Data Service Architecture: A Survey, *Journal of Internet Technology*, Vol. 21, No. 2, pp. 393-405, March, 2020.

[6] J. Leskovec, A. Rajaraman, J. D. Ullman, *Mining of Massive Datasets*, Cambridge University Press, 2014.

[7] C. Li, H. J. Yang, F. Sun, J. M. Cioffi, L. Yang, Multiuser Overhearing for Cooperative Two-way Multiantenna Relays, *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 5, pp. 3796-3802, May, 2016.

[8] J. Yin, J. Wang, A Dirichlet Multinomial Mixture Model-based Approach for Short Text Clustering, *20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, 2014, pp. 233-242.

[9] M. Sahami, T. D. Heilman, A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets, *15th International Conference on World Wide Web*, Edinburgh, Scotland, 2006, pp. 377-386.

[10] C. Li, S. Zhang, P. Liu, F. Sun, J. M. Cioffi, L. Yang, Overhearing Protocol Design Exploiting Intercell Interference in Cooperative Green Networks, *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 1, pp. 441-446, January, 2016.

[11] S. He, Z. Li, Y. Tang, Z. Liao, F. Li, S.-J. Lim, Parameters Compressing in Deep Learning, *Computers, Materials & Continua*, Vol. 62, No. 1, pp. 321-336, 2020. DOI: 10.32604/cmc.2020.06130.

[12] M. Kusner, Y. Sun, N. Kolkin, K. Weinberger, From Word Embeddings to Document Distances, *Conference on Machine Learning,* Lille, France, 2015, pp. 957-966.

[13] U. Manber, Finding Similar Files in a Large File System, *USENIX Winter 1994 Technical Conference*, San Francisco, USA, 1994, pp. 1-10.

[14] C. Zhou, Y. Lu, L. Zou, R. Hu, FICW: Frequent Itemset Based Text Clustering with Window Constraint, *Wuhan University Journal of Natural Sciences*, Vol. 11, No. 5, pp. 1345-1351, September, 2006.

[15] C. Li, F. Sun, J. M. Cioffi, L. Yang, Energy Efficient MIMO Relay Transmissions Via Joint Power Allocations, *IEEE Transactions on Circuits and Systems*, Vol. 61, No. 7, pp. 531-535, July, 2014.

[16] X. Liu, P. He, A Study on Text Clustering Algorithms Based on Frequent Term Sets, in: X. Li, S. Wang, Z. Y. Dong (Eds.), *Advanced Data Mining and Applications. ADMA 2005. Lecture Notes in Computer Science*, Vol. 3584, Springer, 2005, pp. 347-354.

[17] L. Wang, L. Tian, Y. Jia, W. Han, A Hybrid Algorithm for Web Document Clustering Based on Frequent Term Sets and k-Means, in: K. C.-C. Chang, W. Wang, L. Chen, C. A. Ellis,

C.-H. Hsu, A. C. Tsoi, H. Wang (Eds.), *Advances in Web and Network Technologies, and Information Management. APWeb 2007, WAIM 2007. Lecture Notes in Computer Science*, Vol. 4537, Springer, 2007, pp. 198-203.

[18] C. Li, P. Liu, C. Zou, F. Sun, J. M. Cioffi, L. Yang, Spectral-efficient Cellular Communications with Coexistent One- and Two-hop Transmissions, *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 8, pp. 6765-6772, August, 2016.

[19] Z. Su, W. Song, M. Lin, J. Li, Web Text Clustering for Personalized E-learning Based on Maximal Frequent Itemsets, *2008 International Conference on Computer Science and Software Engineering*, Wuhan, China, 2008, pp. 452-455.

[20] Y. Wang, Y. Jia, S. Yang, Short Documents Clustering in Very Large Text Databases, in: L. Feng, G. Wang, C. Zeng, R. Huang (Eds.), *Web Information Systems – WISE 2006 Workshops. WISE 2006. Lecture Notes in Computer Science,* Vol. 4256, Springer, 2006, pp. 83-93.

[21] W. Liu, X. Zheng, Documents Clustering Based on Frequent Term Sets, *Eighth IASTED International Conference on Intelligent Systems and Control*, Cambridge, MA, USA, 2005, pp. 277-280.

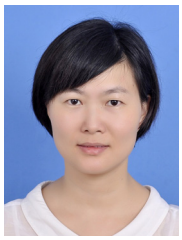
[22] H. Anaya-Sánchez, A. Pons-Porrata, R. Berlanga-Llavori, A Document Clustering Algorithm for Discovering and Describing Topics, *Pattern Recognition Letters*, Vol. 31, No. 6, pp. 502-510, April, 2010.

[23] M. Datar, N. Immorlica, P. Indyk, V. S. Mirrokni, Locality-sensitive Hashing Scheme Based on P-stable Distributions, *Symposium on Computational Geometry*, Brooklyn, New York, USA, 2004, pp. 253-262.

[24] C. Li, H. J. Yang, F. Sun, J. M. Cioffi, L. Yang, Adaptive Overhearing in Two-way Multi-antenna Relay Channels, *IEEE Signal Processing Letters*, Vol. 23, No. 1, pp. 117-120, January, 2016.

[25] J. Wang, Y. Tang, S. He, C. Zhao, P. K. Sharma, O. Alfarraj, A. Tolba, LogEvent2vec: LogEvent-to-Vector Based Anomaly Detection for Large-Scale Logs in Internet of Things, *Sensors,* Vol. 20, No. 9, pp. 2451, May, 2020.

Biographies



Min Zhu received B.S. degree from Nanjing University of Posts and Telecommunications, China in 2002, M.S. degree from Beijing University of Posts and Telecommunications, China in 2005, and received Ph.D. degree in Nanjing University of Posts and Telecommunications, China in 2018. Now, she works at College of Information Science and Technology, Zhejiang Shuren University as lecturer. Her research interests mainly include routing protocol and optimization algorithm design.

Xin Xia received his B.S. and M.S. degrees in Computer Science and Technology from Wuhan University, Wuhan, China, in 2003 and 2006, respectively. From 2010 to 2013, he worked for ZTE Corp. as a senior pre-research engineer. From 2013 to 2019, he has been with the Department of Information Technology, Wenzhou Vocational & Technical College, Wenzhou, China. He is currently working with the College of Computer Information Technology, Wuhan Institute of Shipbuilding Technology, Wuhan, China. His research interests include computer network and communication, machine learning, and computer vision.

Jianming Zhang received the B.S. and M.S. degree in 1996 and 2001 respectively from Zhejiang University and the National University of Defense Technology, China. He received the Ph.D. in 2010 from Hunan University, China. Currently, he is a professor and the deputy dean in the School of Computer and Communication Engineering at Changsha University of Science and Technology, China. His main research interests lie in the areas of computer vision, data mining, and wireless ad hoc & sensor networks. He has published more than 70 research papers. He is a member of IEEE and a senior member of CCF.

Dengyin Zhang received the B.S., M.S., and Ph.D. degrees from Nanjing University of Posts and Telecommunication, Nanjing, China, in 1986, 1989, and 2004, respectively. He is currently a Professor of the School of Internet of Things, Nanjing University of Posts and Telecommunication, Nanjing, China. He was a visiting scholar in Digital Media Lab, Umea University, Sweden, from 2007 to 2008. His research interests include signal and information processing, networking technique, and information security.