Dask-Based Efficient Clustering of Educational Texts*

Fail Gafarov[1][0000−0003−4704−154X], Dmitriy Minullin[1][0000−0001−7713−5251], and Viliuza Gafarova[2][0000−0001−7215−4007]

[1] Kazan Federal University, 18 Kremlyovskaya street. Kazan 420008, Russian Federation fgafarov@yandex.ru
[2] TR AS Institute of Applied Semiotics Levo-Bulachnaya str., 36A, Kazan, 420111, Russian Federation

Abstract. Document clustering process is a long running and computationally demanding process. The need for systems that allow fast document clustering is especially relevant for processing large volumes of text data (Big Data). In this work we present a distributed text clustering framework based on Dask open source library for parallel and distributed computing. The Dask-based processing system developed in this work allows to execute all necessary operations related to the clustering of text documents in a parallel mode. We realized parallel agglomerative clustering algorithm of cosine similarity matrices computed from term frequency-inverse document frequency (TF-IDF) feature matrices of input texts. The system had been applied to intellectual analysis of educational data accumulated in the system "Electronic education of the Tatarstan Republic" from 2015 to 2020. Specially, by using developed system we clustered the text documents describing lesson planning, and also performed a comparative analysis of the average marks of students, whose training was carried out according to lesson planning belonging to different clusters.

Keywords: Big Data, Dask, educational data mining, python, document clustering, TF-IDF, ANOVA

1 Introduction

Big data analytics describes the methods of processing large amounts of data in order to discover hidden patterns [1], market trends, customer preferences and other useful information for making the right decisions [11], [30]. It has been adopted by a wide variety of industries and has become a separate industry [26]. Using the Big Data methods in educational system includes measuring, collecting, analyzing and presenting huge volumes of structured and unstructured data about students and the educational environment in order to understand the peculiarities of the functioning and development of the educational system [25], [12]. For the successful management of the educational process, it is necessary

* The reported study was funded by RFBR, project number 19-29-14082

to promptly process numerous various incoming data online, so the use of Big Data technologies becomes a necessity [19]. Working with large amounts of data requires not only the availability of modern hardware, but also mathematical algorithms that would reduce the required number of computing operations for a computer [3], [37].

Text mining in big data analytics is emerging powerful tool for the analysis of unstructured textual data. These methods are used to extract new knowledge and to identify significant patterns and correlations hidden in the data [16]. A method for extracting associative feature information by using text mining from health big data is proposed in [21], social network analysis algorithms are utilized for identifying the emerging trends for big data domain [18], text mining has gained its popularity with big data resources when analyzing big data in the financial sector [6].

Document clustering is the process of finding groups of similar documents in a collection of documents. Clustering algorithms are among the most popular data mining methods, and are widely used for processing text data. They have a wide range of applications such as classification [5, 4], visualization [8] and organization of documents [13]. Various methods of text clustering are used, the most popular of which are LSA / LSI - Latent Semantic Analysis / Indexing [10], Suffix Tree Clustering [44], Scatter/Gather [9]. Recently, methods based on the use of neural networks have gained popularity [27] together with classical clustering methods, such as the k-means algorithm [28].

In document clustering tasks the problem of determining the optimal number of clusters arises [23], [36]. By using hierarchical clustering, researchers often have to work with dendrograms and manually analyze them and trim the required number of clusters. This approach is followed by several problems, firstly, this is the obvious slow nature of the study, secondly, it is the human factor of the possibility of making mistakes. In cases where the threshold is not easy to determine, two different researchers may come to different conclusions about the correct number of clusters. There are various approaches for determining the optimal number of clusters: gap statistic, elbow method, mode, maximum difference [43], [20], [45].

This work demonstrates the possibilities of using the Dask cluster computing system [32] for analyzing the texts describing lesson planning in schools. The study is based on the data collected in the state information system "Electronic education in the Republic of Tatarstan." This system includes a large-scale database of educational information on all students and all teachers of schools located in Tatarstan Republic. Teachers fill out lesson topics in the electronic journal, as well as homework through their personal accounts. They also fill in the system and current grades given to their students.

Be believe that the success of educational process is depending on the teaching material used by teacher. The main goal of this work is aimed to build an analytical system for finding and studying the differences between the academic performance of schoolchildren studying in different educational and methodological complexes or in different curricula. We can obtain information about

the differences in teaching and methodological materials only on the basis of an analysis of textual data on the conducted lessons, which were filled by teachers through their personal accounts. The main problem is that these data are not marked up and not annotated, i.e. teachers do not indicate which curriculum they teach their lessons. Therefore to solve this problem, we need to use an automatic text clustering system.

For this research depersonalized datasets were provided in Comma-separated values (CSV files) format. These files contains information about teachers, pupils, lessons and marks for all subjects and all grades. The total amount of data is over 60 GB. Because the original data consists of a large number of texts (Big Data) of different sizes that will take a lot of time to process by using traditional methods, we need to develop high-performance computing (HPC) programming based on cluster computing to efficiently processing text data.

In this work we developed HPC computational framework based on Dask distributed library [32]. To convert a document into structured format the weighting schema TF-IDF (Term Frequency –Inverse Document Frequency) is used. To measure the distance between the text we used cosine similarity algorithm. The division of texts into clusters was carried out by using an Agglomerative clustering algorithm, with the help of Elbow method to find the optimal number of clusters. Bu using the results of text clustering, we carried out a comparative analysis of the average academic performance of students whose lesson planning texts belong to different clusters. The developed system makes it possible to efficiently and quickly carry out a full-fledged analysis of this type for all grades and subject.

2 Dask-Based Parallel Processing Framework

To efficiently and quickly process a large amount of unstructured data we have to use BigData technologies, and also the powers of computing clusters are required. For high-performance calculations a computing cluster containing 4 virtual machines was deployed (each VM has 1TB HDD, 32 GB RAM, 16 CPU cores), with installed Python-based library for parallel computing - Dask. Dask is a flexible parallel big data processing library, designed to provide scalability and to extend the capabilities of existing Python packages and libraries [32]. Dask allows users to integrate and to run in parallel mode existing Python-based scripts written by using popular libraries such as NumPy, SciPy, Pandas and others. The main advantage of using the Dask library for processing large texts is based on the ability of the system to perform computations with data volumes that are larger than the available memory of single computer [39], [17], [14].

The functionality of Dask which is necessary to perform its tasks can be divided into two parts:

1. Dynamic task scheduling optimized for cluster based HPC computation.
2. “Big Data” collections like parallel arrays, dataframes, and lists that extend common interfaces like NumPy, Pandas,or Python iterators to larger-than-

memory or distributed environments. These parallel collections run on top of dynamic task schedulers.

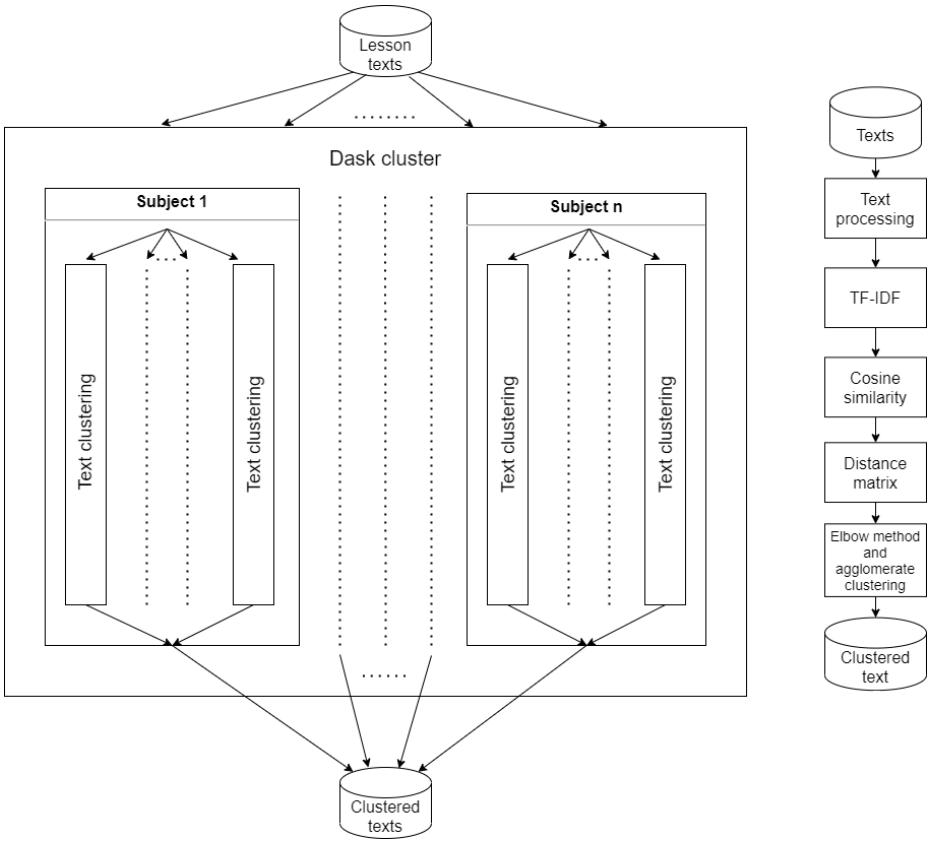

Fig. 1. Schematic representation of the system's architecture

All data, initially obtained in the form of ssv files, had been combined into more convenient data structures (DataFrames) by using Dask's *merge()* method, and stored on cluster's computer's hard drives in *parquet* binary data format [42]. This dataframe, in addition to the texts describing the topics of the lessons, also contains an information about lesson's dates, grades, subjects and teachers. The distributed data processing framework is schematically drawn in Figure 1 in the left panel. In the the right side of the Figure 1 sequential steps of the text processing pipeline, executed in parallel mode on the computational cluster are demonstrated. In the first processing step, we pre-processed the entire text corpus to remove noisy and less useful words. Next, we applied TF-IDF and cosine similarity calculation, followed by hierarchical agglomerative clustering, by using

the "elbow" method to determine the number of clusters. Such parallel execution of processing methods is started by calling dataframe's *apply* method, and by specifying the corresponding method name, which must be executed in parallel mode as a parameter of dataframe's *apply* method. To perform parallel processing of school subjects (*ProcSubjects*), the following call to the *apply* method is used:

```
lessons_texts.groupby(['subjectID']).apply(ProcSubjects).compute()
```

The method for processing texts for particular grades (*ProcessGrades*) was launched in a similar way:

```
def ProcSubjects(df):
    df.groupby(['grade_number']).apply(ProcessGrades)
```

Data reprocessing. As the first step the texts (originally there were 95786285 texts), describing lessons, were combined into a 580698 text documents, for each subject and for each teacher. These texts had been subjected to the following processing:

1. The document texts are divided into tokens
2. From the array of tokens we removed punctuation marks, empty lines and stop words, which do not have any special meaning to the sentence.

Theses steps are necessary to improve the accuracy of the text's comparison.

Document text vectorization by term frequency-inverse document frequency (TF-IDF) method. Traditional text similarity measurements use TF-IDF (term frequency (TF)x inverse document frequency (IDF)) method (first introduced in [35]) to compute similarity between text documents by using cosine similarity [7], to examine the relevance of words to documents [31], short-text clustering [38], text categorization [41], pattern mining on text data[2]. TF-IDF reflects how important a word is to a document in a collection or corpus, and therefore it is often used also as a weighting factor in information retrieval and text mining. The TF-IDF value is proportional to how often a word occurs in a particular document, but is compensated for by the frequency of the word in the entire corpus. This property of TF-IDF helps to take into account the fact that some words are usually much more common than others [7].

For a collection of words $t \in T$ that appears in a set of N documents $d \in D$ with length n_d, IF-IDF is computed by the formulas

$$TF = \frac{f_{t,d}}{n_d} \tag{1}$$

$$IDF = \log \frac{N}{df_t} \tag{2}$$

$$W = TF * IDF, \tag{3}$$

where $f_{t,d}$ is the frequency of word t in document d, df_t is the number of documents in which the word t appears.

Cosine similarity and distance matrix. At the next step we constructed a text similarity matrix by making similarity analysis of the compared texts. The cosine similarity method [40] was chosen as a comparison method for TF-IDF vectorized texts. Cosine similarity is a measure of the similarity between two vectors, and measures the cosine of the angle between them. For two feature vectors, a and b, cosine similarity, $\cos(\theta)$, can be represented using the dot product and the [15] norm:

$$\cos(\theta) = \frac{\boldsymbol{ab}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i}\sqrt{\sum_{i=1}^{n} b_i}}, \tag{4}$$

$\boldsymbol{a}$ – coordinates of the first vector (TF-IDF vector for first text), $\boldsymbol{b}$ – coordinates of the second vector (TF-IDF vector for the second text). Further, calculating cosine similarity in pairs of vectors by using a cyclic algorithm a text similarity matrix is constructed. Based on the similarity matrix, we calculated the distance matrix (by formula $distance\_matrix = 1 - similarity\_matrix$) for using it as an input data for agglomerative clustering algorithm.

Agglomerative clustering. Clustering is one of the most common unsupervised machine learning problems. For text clustering we used the most common type of hierarchical clustering - the agglomerative clustering. This method is used to group objects in clusters based on their similarity. The algorithm starts by treating each object as a single cluster. In the next steps, pairs of clusters are successively merged until all clusters have been merged into one big cluster containing all objects [33]. The results of agglomerative clustering are usually presented in a dendrogram (an example of dendrogramm is presented in Figure 2).

The dendrogram is convenient, because it allows to visually observe the clustering process. To calculate the distance between clusters Ward's method was used. Ward's minimum variance method calculates the distance between cluster members and the centroid. The centroid of a cluster is defined as the point at which the sum of squared Euclidean distances between the point itself and each other point in the cluster is minimised. The increment in the sum of the squares of the distances of objects to the center of the cluster, obtained as a result of their union, is taken as the distance between the clusters As a result, we get a complete clustering tree of our initial set from which we can get clustering of any level [29].

Determining the number of clusters by "elbow" method. Determining the optimal number of clusters is a fundamental issue in partitioning clustering, which requires the user to specify the number of clusters k to be generated. A simple and popular method consists of visual analysis of the dendrogram to see if it suggests a particular number of clusters. This approach is very subjective, but its main drawback is the impossibility of visual analysis of hundreds of dengrograms within the framework of automatic processing. To find optimal number of clusters in automatic mode we use "elbow" method. This method is based on calculating the Within-Cluster-Sum of Squared Errors (WSS) for different number of clusters (k) and selecting the k for which change in WSS starts

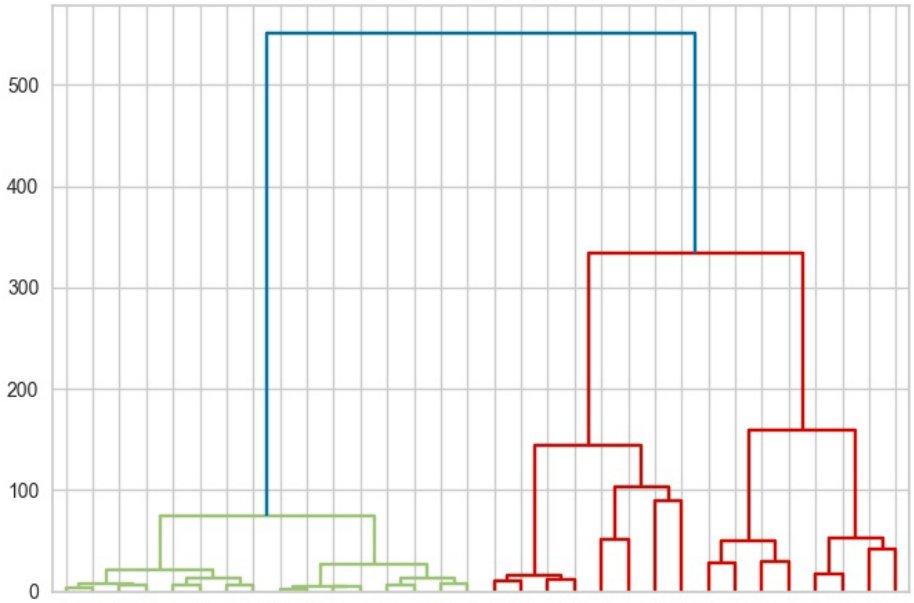

Fig. 2. Example of dendrogram for subject mathematics in 3rd grade

to diminish. Based on the numerical results carried out in the [43] elbow method and the maximum difference seem to perfectly capture the simulated data.

To demonstrate "elbow" method we plot a line chart of the SSE for each value of k (see Figure 3). Our goal is to choose a small value of k that still has a low SSE, and the elbow point usually represents the point where we start to have diminishing returns by increasing k (Figure 3). We used the *KElbowVisualizer* class from *yellowbrick* Python library to select the optimal number of clusters.

Now, knowing the number of clusters by using the "eblow" method, we can split our set of texts into k clusters. In figure 4 we show the numbers of texts by clusters for subject "mathematics" in 3-th grade.

For checking the reliability of the results obtained using the cluster-based computational system based on Dask, as well as to assess the performance of parallel cluster processing pipeline, we have performed the same data computations by using desktop computer without using the Dask system. It should be noted that during performing the same processing text processing pipelines on usual desktop computers we we ran into problems, mainly due to the fact that processing large data files requires large amounts of RAM. To solve this problem we need to split huge text files into smaller files (which must be small enough to fit into the RAM of a desktop computer), and process them sequentially, constantly reading and writing intermediate results to hard drives, which slows down data processing a dozen times. Data processing, which takes about an hour by using a Dask-based cluster system (4 VMs with 1TB HDD, 32 GB RAM, 16 CPU cores), tooks a several days on desktop computers.

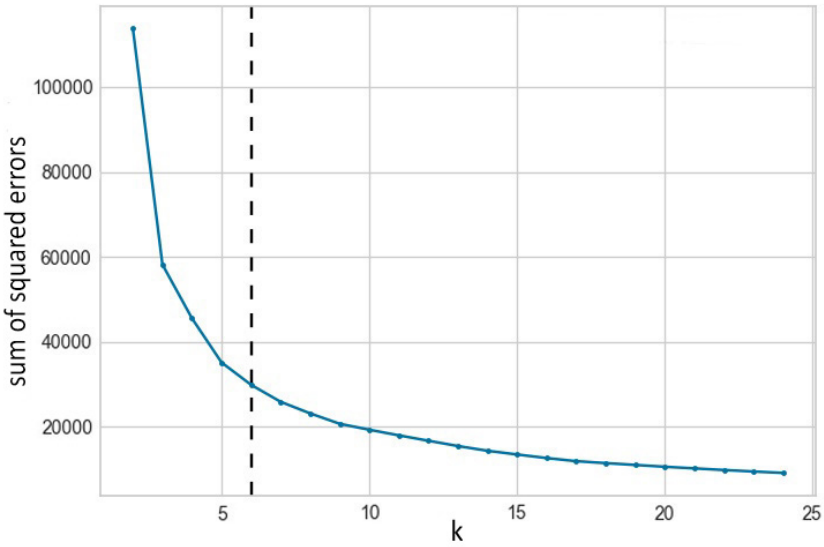

Fig. 3. Exaple of "elbow" end point for subject "mathematics" in 3th grade

3 Results

By using the computational framework, presented above, we performed cluster analysis of the text describing lesson's for all school subjects for all grades (2-11). To confirm the reliability of the clustering results obtained by our system, some of the texts (mathematics, 3rd grade) were by-hand analyzed by teachers, who confirmed the correctness of automatic division into clusters. These teachers also confirmed that our system correctly identifies educational and methodological complexes that are used in primary grades. We estimated that the accuracy of correctly clustered text on manually tested subset achieved 93%. Incorrectly clustered texts (7%) is due to the fact that some teachers did not completely correctly and incompletely entered information into the system.

A comparative analysis of WSS for mathematics and Russian language subjects among grades 2-11 (Fig. 5), shows that in the primary grades the values of WSS are very high, what means a presence strong difference between lesson texts. This difference gradually disappears by 10-11 class. A similar picture is observed for all academic years (in the Figure 5, we presented the results only for the 2015-2016 and 2016-2017 academic years). This is due to the fact that after primary school in the middle level school, students learns according to a certain program. The almost identical similarity of the lines in the 10th and 11th grade is because that all students in all schools follow the same curriculum.

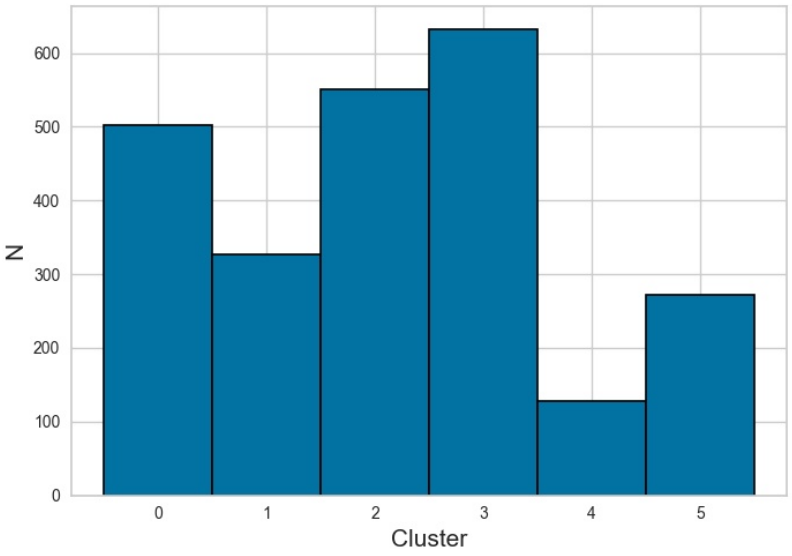

Fig. 4. The number of texts in each cluster for the texts of lessons on the subject of mathematics in 3-th grade

These features discovered by our system are confirmed by teachers who work in schools, which confirms the reliability of our conclusions.

Since there is a significant difference between the texts of lesson planning in primary grades, we conducted a study of the dependence of the average grades of students on clusters of lesson planning texts. For the primary preprocessing of student's marks, which consisted in grouping data and calculating the average marks of students we also used capabilities of the Dask library. As a result, we get the following marks (Figure 6), on the basis of which the analysis of variance will be carried out. Analysis of variance (ANOVA) was used to find and to determine the differences between the mean marks of students belonging to different clusters[22]. ANOVA-test had been used because it allows simultaneous comparison of a larger number of samples, in contrast to Student's t-test, which allows only pairwise comparison. The results of this analysis do not show the exact differences between groups, but can only tell where they are and where they are not. In other words, if the hypothesis of the equality of all groups is rejected, this does not mean that all groups are different, but only that some of the groups are different, maybe everything, but maybe only a few.

The p-value is calculated by using the sampling distribution of the test statistic under the null hypothesis, the sample data, and the type of test being done (lower-tailed test, upper-tailed test, or two-sided test). After the analysis for

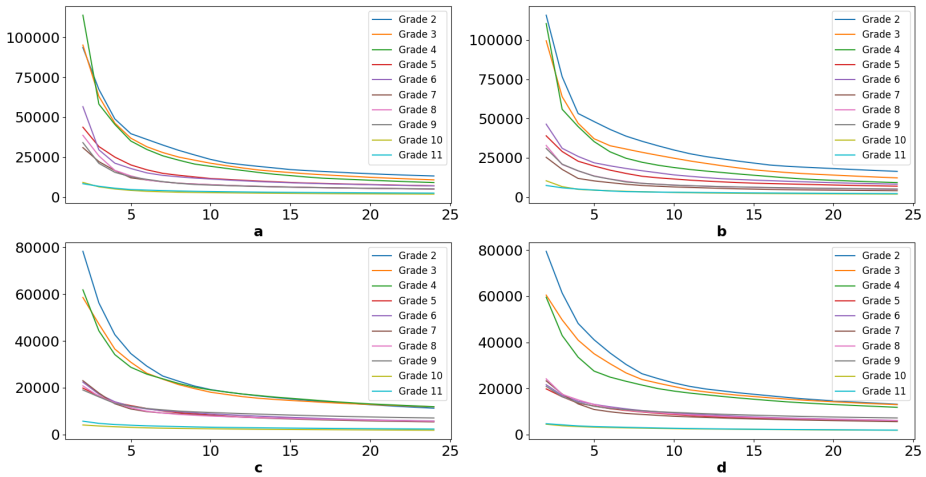

Fig. 5. WSS for different values of clusters count k (a - mathematics 2015-2016, b - mathematics 2016-2017, c - Russian language 2015-2016, d - Russian language 2016-2017.)

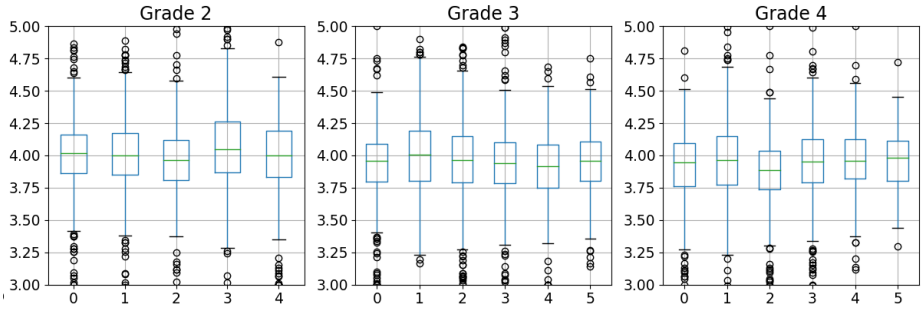

Fig. 6. Box-plots of student's mean mark by cluster in mathematics for 2015-2016 academic

different subject for primary school grades (2-4) we get the following values of the p-value by grades and subjects (Table 1):

Based on the results obtained, for the primary school level, subjects can be divided into two groups :

1. Subjects that do not depend on the educational-methodical complex (p-values > 0.05): music, English language, physical education,
2. Subjects depending on the educational and methodological complex (p-values < 0.05): mathematics, Russian language, literature, technology, arts.

For the middle and senior levels of the school into three groups (data not shown):

	2-nd grade	3-rd grade	4-th grade
Physical education	0,100	0,045	0,041
Mathematics	0,000	0,003	0,000
Russian language	0,000	0,000	0,000
Literature	0,000	0,000	0,000
Technology	0,010	0,009	0,005
Music	0,896	0,901	0,000
Arts	0,000	0,001	0,000
English language	0,726	0,006	0,561

Table 1. One-way ANOVA-test p-values for some primary school subjects for the 2015-2016 academic year

1. Subjects that do not depend on the educational and methodological complex: English language, computer science, technology.
2. Subjects depending on the educational and methodological complex: mathematics, physical education, Russian language, literature, physics, Tatar language, Tatar literature.
3. Floating subjects: biology, geography, history, social studies, chemistry.

Comparing with the results of the other years at the primary school level, this situation remains the same, with the exception of music, which have moved to other groups. For middle and senior school level, the situation remains largely unchanged except for the history subject that moved into the first group.

4 Conclusion

By now, various databases of governmental and non-governmental organizations have been accumulated huge amounts of various types data. Because of a large volumes of datasets, for processing and analysing these datasets Big Data methods are needed. In this work, we have built an information-analytical system for processing data describing in detail the school educational process. The system is built on the basis of the distributed computing Dask library. The use of this system allows to perform efficient and high-performance analysis (clustering) of text data. We demonstrated the application of the developed system for data processing stored in "Electronic Education of the Republic of Tatarstan" system, and have obtained interesting and important conclusions in the educational analytic field. The approach developed in this work can be used not only in education systems, it can be used for any system where it is necessary to conduct cluster analysis of large volumes of text (or non-text) data.

The academic success of schoolchildren depends on many different factors associated with the educational environment, with the individual characteristics of students, and so on. Naturally, the textual data of educational and methodological complexes are not the only the single factor that completely determines the academic success of schoolchildren. But we believe this is one of the most important factors. In this work, on the basis of quantitative measures, we were

have foun that in different classes for different disciplines, educational-methodical complexes have a different level of influence on student performance.

Apache Spark [34] based tools are also quite often used for big data processing. However, Dask is smaller, lighter and simpler than Spark, and flexible as Pandas. Program scripts are easy to debug on personal computers before deploying on a cluster, because Dask also works on personal computers. In our project, we have deployed our system on a static computing cluster, but in future we will use the Kubernetes-like technology [24] for elastic deploying in Amazon or Google clouds.

Acknowledgements. The reported study was funded by RFBR, project number 19-29-14082.

References

1. Adnan, K., Akbar, R.: An analytical study of information extraction from unstructured and multidimensional big data. Journal of Big Data **6**(1), 91 (2019). https://doi.org/10.1186/s40537-019-0254-8
2. Agnihotri, D., Verma, K., Tripathi, P.: Pattern and cluster mining on text data. In: 2014 Fourth International Conference on Communication Systems and Network Technologies. pp. 428–432 (2014). https://doi.org/10.1109/CSNT.2014.92
3. Aljawarneh, S., Yassein, M.B., Talafha, W.A.: A resource-efficient encryption algorithm for multimedia big data. Multimedia Tools and Applications **76**(21), 22703–22724 (2017). https://doi.org/10.1007/s11042-016-4333-y
4. Angelova, R., Siersdorfer, S.: A neighborhood-based approach for clustering of linked document collections. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management. p. 778–779. CIKM '06, Association for Computing Machinery (2006). https://doi.org/10.1145/1183614.1183726
5. Anick, P.G., Vaithyanathan, S.: Exploiting clustering and phrases for context-based information retrieval. SIGIR Forum **31**, 314–323 (1997). https://doi.org/10.1145/278459.258601
6. Bach, M., Krstic, Z., Seljan, S., Turulja, L.: Text mining for big data analysis in financial sector: A literature review. Sustainability (Switzerland) **11**(5) (2019). https://doi.org/10.3390/su11051277
7. Bafna, P., Pramod, D., Vaidya, A.: Document clustering: Tf-idf approach. In: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). pp. 61–66 (2016). https://doi.org/10.1109/ICEEOT.2016.7754750
8. Chakrabarti, K., Mehrotra, S.: Local dimensionality reduction: A new approach to indexing high dimensional spaces. In: Proceedings of the 26th VLDB Conference. pp. 89–100 (2000)
9. Cutting, D.R., Karger, D.R., Pedersen, J.O., Tukey, J.W.: Scatter/gather: A cluster-based approach to browsing large document collections. In: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 318–329. SIGIR '92, Association for Computing Machinery (1992). https://doi.org/10.1145/133160.133214
10. Dumais, S.: Latent semantic analysis. Annual Review of Information Science and Technology **38**, 188–230 (2004). https://doi.org/10.1002/aris.1440380105

11. Elgendy, N., Elragal, A.: Big data analytics: A literature review paper. In: Perner, P. (ed.) Advances in Data Mining. Applications and Theoretical Aspects. pp. 214–227. Springer International Publishing, Cham (2014)
12. Elia, G., Solazzo, G., Lorenzo, G., Passiante, G.: Assessing learners' satisfaction in collaborative online courses through a big data approach. Computers in Human Behavior **92**, 589–599 (2019). https://doi.org/https://doi.org/10.1016/j.chb.2018.04.033
13. Fisher, D.: Knowledge acquisition via incremental conceptual clustering. Machine Learning **2**, 139–172 (1987). https://doi.org/10.1023/A:1022852608280
14. Ford, A.S., Weitzner, B.D., Bahl, C.D.: Integration of the rosetta suite with the python software stack via reproducible packaging and core programming interfaces for distributed simulation. Protein Science **29**(1), 43–51 (2020). https://doi.org/https://doi.org/10.1002/pro.3721
15. Grishin, V.: Method of analysis and search for borrowings in the text. Problems of science **7**, 31 (2018)
16. Hassani, H., Beneki, C., Unger, S., Mazinani, M.T., Yeganegi, M.R.: Text mining in big data analytics. Big Data and Cognitive Computing **4**(1) (2020). https://doi.org/10.3390/bdcc4010001
17. Henriques, J., Caldeira, F., Cruz, T., Simões, P.: Combining k-means and xgboost models for anomaly detection using log datasets. Electronics **9**(7) (2020). https://doi.org/10.3390/electronics9071164
18. Jalali, S.M.J., Park, H.W., Vanani, I.R., Pho, K.H.: Research trends on big data domain using text mining algorithms. Digital Scholarship in the Humanities (04 2020). https://doi.org/10.1093/llc/fqaa012
19. Javidi, G., Rajabion, L., Sheybani, E.: Educational data mining and learning analytics: Overview of benefits and challenges. In: 2017 International Conference on Computational Science and Computational Intelligence (CSCI). pp. 1102–1107 (2017). https://doi.org/10.1109/CSCI.2017.360
20. Jung, Y., Park, H., Du, D.Z., Drake, B.L.: A decision criterion for the optimal number of clusters in hierarchical clustering. Journal of Global Optimization **25**(1), 91–111 (2003). https://doi.org/10.1023/A:1021394316112
21. Kim, J.C., Chung, K.: Associative feature information extraction using text mining from health big data. Wireless Personal Communications **105**(2), 691–707 (Mar 2019). https://doi.org/10.1007/s11277-018-5722-5
22. Kim, T.: Understanding one-way anova using conceptual figures. Korean Journal of Anesthesiology **70**, 22 (02 2017). https://doi.org/10.4097/kjae.2017.70.1.22
23. Kothari, R., Pitts, D.: On finding the number of clusters. Pattern Recognition Letters **20**(4), 405–416 (1999). https://doi.org/https://doi.org/10.1016/S0167-8655(99)00008-2
24. Kristiani, E., Yang, C.T., Wang, Y.T., Huang, C.Y.: Implementation of an edge computing architecture using openstack and kubernetes. In: Kim, K.J., Baek, N. (eds.) Information Science and Applications 2018. pp. 675–685. Springer Singapore, Singapore (2019)
25. Logica, B., Magdalena, R.: Using big data in the academic environment. Procedia Economics and Finance **33**, 277–286 (2015). https://doi.org/https://doi.org/10.1016/S2212-5671(15)01712-8
26. Lu, Y.: Industry 4.0: A survey on technologies, applications and open research issues. Journal of Industrial Information Integration **6**, 1–10 (2017). https://doi.org/https://doi.org/10.1016/j.jii.2017.04.005

27. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings (2013), http://arxiv.org/abs/1301.3781
28. Mogotsi, I.: Christopher d. manning, prabhakar raghavan, and hinrich schütze: Introduction to information retrieval. Inf. Retr. **13**, 192–195 (2010). https://doi.org/10.1007/s10791-009-9115-y
29. Murtagh, F., Legendre, P.: Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? Journal of Classification **31**(3), 274–295 (2014). https://doi.org/10.1007/s00357-014-9161-z
30. Poleto, T., de Carvalho, V.D.H., Costa, A.P.C.S.: The roles of big data in the decision-support process: An empirical investigation. In: Delibašić, B., Hernández, J.E., Papathanasiou, J., Dargam, F., Zaraté, P., Ribeiro, R., Liu, S., Linden, I. (eds.) Decision Support Systems V – Big Data Analytics for Decision Making. pp. 10–21. Springer International Publishing, Cham (2015)
31. Qaiser, S., Ali, R.: Text mining: Use of tf-idf to examine the relevance of words to documents. International Journal of Computer Applications **181**(1), 25–29 (Jul 2018). https://doi.org/10.5120/ijca2018917395
32. Rocklin, M.: Dask: Parallel computation with blocked algorithms and task scheduling. In: Python in Science Conference. pp. 126–132 (2015). https://doi.org/10.25080/Majora-7b98e3ed-013
33. Roux, M.: A comparative study of divisive and agglomerative hierarchical clustering algorithms. Journal of Classification **35**(2), 345–366 (2018). https://doi.org/10.1007/s00357-018-9259-9
34. Salloum, S., Dautov, R., Chen, X., Peng, P.X., Huang, J.Z.: Big data analytics on apache spark. International Journal of Data Science and Analytics **1**(3), 145–164 (Nov 2016). https://doi.org/10.1007/s41060-016-0027-9
35. Salton, G., Yang, C.: On the specification of term values in automatic indexing. Journal of Documentation **29**, 351–372 (1973)
36. Salvador, S., Chan, P.: Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: 16th IEEE International Conference on Tools with Artificial Intelligence. pp. 576–584 (2004). https://doi.org/10.1109/ICTAI.2004.50
37. Scott, S.L., Blocker, A.W., Bonassi, F.V., Chipman, H.A., George, E.I., McCulloch, R.E.: Bayes and big data: the consensus monte carlo algorithm. International Journal of Management Science and Engineering Management **11**(2), 78–88 (2016). https://doi.org/10.1080/17509653.2016.1142191
38. Seifzadeh, S., Farahat, A.K., Kamel, M.S., Karray, F.: Short-text clustering using statistical semantics. In: Proceedings of the 24th International Conference on World Wide Web. p. 805–810. WWW '15 Companion, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2740908.2742474
39. Signell, R.P., Pothina, D.: Analysis and visualization of coastal ocean model data in the cloud. Journal of Marine Science and Engineering **7**(4) (2019). https://doi.org/10.3390/jmse7040110
40. Singhal, A.: Modern information retrieval: a brief overview. IEEE Data Engineering Bulletin **24** (2001)
41. Trstenjak, B., Mikac, S., Donko, D.: Knn with tf-idf based framework for text categorization. Procedia Engineering **69**, 1356–1364 (2014). https://doi.org/https://doi.org/10.1016/j.proeng.2014.03.129, 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013

42. Vohra, D.: Apache Parquet, pp. 325–335. Apress, Berkeley, CA (2016)
43. Zambelli, A.: A data-driven approach to estimating the number of clusters in hierarchical clustering. F1000Research **5** (2016). https://doi.org/10.12688/f1000research.10103.1
44. Zamir, O., Etzioni, O.: Web document clustering: A feasibility demonstration. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 46–54. SIGIR '98, Association for Computing Machinery, New York, NY, USA (1998). https://doi.org/10.1145/290941.290956
45. Zhou, S., Xu, Z., Liu, F.: Method for determining the optimal number of clusters based on agglomerative hierarchical clustering. IEEE Transactions on Neural Networks and Learning Systems **28**(12), 3007–3017 (2017). https://doi.org/10.1109/TNNLS.2016.2608001