• ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ

УДК: 004.387 /МРНТИ: 83.77.33 https://doi.org/10.51301/vest.su.2021.i3.06

M. Meraliyev, K. Orynbekova, A. Talasbek*, D. Kariboz, A. Issabek
Suleyman Demirel University, Kazakhstan
*e-mail: assem.talasbek@sdu.edu.kz

OPTIMIZATION OF DATA SEGMENTS AND NUMBER OF CORES FOR DEFINING POPULARITY OF KAZAKH WORDS USING APACHE SPARK

Abstract. Kazakh is an agglutinative language which has complex structure. In this work Apache Spark was used to specify the popularity of Kazakh words in 3 popular kazakh compositions. The main goal was to find the optimal number of data segments for a specific number of cores in order to find the best computational speed. To do so, the data was divided into several segments and ran on a cluster with a different number of cores each time. Results show that the amount of data segments directly affects the computing speed.

Keywords: Apache Spark, RDD, data partitions, NLP, MapReduce paradigm.

Introduction. Kazakh literature, tightly interwoven with traditions, goes back to centuries. Today, our literature preserves national identity and opens up the philosopher Al-Farabi, the thinker Abay, the classic Mukhtar Auezov to the world. Through the work of writers, the world community learns the spiritual world of the nation.

In this project we determined popular words used in the outstanding pieces of art in Kazakh language such as the multivolume novel of M.Auezov "The path of Abay" ("Abay Zholy"), the trilogy of I. Esenberlin "The nomads" ("Koshpediler") and roman about T. Ryskulov of Sh. Murtaza " The Red Arrow" ("Kyzyl Zhebe") translated into many languages of the world.

Kazakh is an agglutinative language that joins morphemes to form long words and belongs to the Ural-Altaic language family. The way of generation of lexical forms in Kazakh language is different from other languages and its structure is complex because of derivational morphology [1, 2]. Also Kazakh language has a vowel harmony rule called synharmonism, which means each syllable should match according to the front or back vowel [3]. All these specifics of agglutinative language complicate the problem of processing it.

Kazakhstan is going to move from Cyrillic to Latin in several years, according to strategic programm "Рухани жаңғыру"[4]. But there are about 20 variants of the new Latin alphabet at the design stage. To choose the most optimal variants of all letters, especially critical letters, authors are going to collect "The path of Abay", intuitively rewritten Latin text by citizens. To get more exact results authors are planning to interrogate more people. To analyze collected data, they chose the Hadoop Distributed System framework, which's processing part working on MapReduce paradigm which is used for large-scale data [5].

According to a huge amount of data, scientists have to use appropriate frameworks and tools. One team from Suleyman Demirel University did sentiment analysis of a Kazakh language via Spark [6]. Apache Spark is easy to use and more flexible for complex problem processing [7]. Therefore, in this project Spark is used.

In one of our previous work, we have shown that using Spark makes it almost three times faster for computing face embeddings even with small amounts of cores [8]. Using a similar approach, in this paper our goal is to compare by how much scale Spark affects computing speed.

Implementation. The process of project development can be splitted into 3 parts. Data Collection, Data Processing and Analysis of ready data. Let us describe each step in more details.

To gather the data for the project we have downloaded books of three grandiose authors: Mukhtar Auezov, Ilyas Esenberlin, Sherkhan Murtaza. As a text we took grandiose books Abay Zholy, Koshpendiler and Kyzyl Zhebe. The goal was to collect more information to get a more exact result. All data will be stored in collections data structures, which will then be used to perform analysis.

The second and most important step of this work is to preprocess the data. In this step we took text from books and started from converting all texts to lowercase in order to overcome duplications. After that we have to remove all punctuation signs and to do so, we have used nltk library. Our next step was to extract separate words using word_tokenizer function form nltk. The last step in preprocessing step was to remove stop words. Since there are no stop words for Kazakh language built in the nltk library, we used to work with the philology department of our faculty to create a collection of stop words for Kazakh language [9, 10].

As a third step in an experiment we used python built-in functions to calculate how many times each word was repeated in text in order to extract the most popular words from these books.

Another experiment was implemented using Pandas, which is an open source powerful tool for data analysis. So we have created a dataframe which consists of words, after that work on calculation of most popular words were done by data aggregation. We have grouped data according to the same words and calculated number of repetitions of each word [11].

In order to increase computation speed Apache Spark was used. Spark is an open source distributed cluster computing system. It works with Resilient Distributed Dataset (RDD) which makes it fault tolerant. It uses a lazy approach, it starts computing only when there is a request to get some data. Spark, like Hadoop, is based on the MapReduce paradigm, but unlike Hadoop, Spark does all the computations in memory and store results there too, which makes it much faster than storing in hard disk [12, 13].

Spark allows not only to engage all the cores in different Namenodes, but also it is possible to specify how many cores per node you want to engage. By selecting a different number of cores in experiments, there is an opportunity to test by how much every single core affects the computational time.

Also in Spark, the amount of segmentations can be given manually, i.e. it is possible to choose how many segments the data must be divided into. Basically, if there are 3 cores and 12 data segments, each core will work with 4 segments on average, and if there are the same 3 cores but only 2 data segments, 2 cores will receive one data segment each, and one core will do nothing. So, to get the best results, it is better to have *n* cores and *m* amount of segments, so that *n* is divisible by *m* [14, 15].

In our experiment we divided our data into 12, 36 and 54 segments and ran them in 6, 12 and 18 cores. Table 1 shows increase of speed with different number of data segments and cores relative to the speed run on 6 cores with data divided into 12 segments in percentage.

As shown in Table 1, the change in speed is significant between 6 and 12 cores, but not so high between 12 and 18 cores. Also, the best results were acquired when the number of segments was divisible by the amount of cores, and the more segments we have, the better results we get.

Table 1. **Computation speed comparison with different amounts of cores and segments.**

Number of cores	Number of data partitions	Increase in speed (in percentage)
6	12	100%
12	12	224%
18	12	255%
6	36	108%
12	36	233%
18	36	262%
6	54	118%
12	54	215%
18	54	289%

Conclusion. A massive increase in information on a daily basis can possibly be seen which means that algorithms that can be performed on a single computer are not a reasonable norm and clustered ones are required. In this research, we're proposing an algorithm to calculate most popular words in a fusional language that operates in a distributed mode using Spark. This work shows that this technology can optimize the process of words calculation, and increase efficiency by decreasing the time. It can be achieved not only by adding more cores, but also by dividing the data into a reasonable number of segments which is directly related to the number of cores. In the future we plan to make an analysis of each word to make this algorithm more accurate to follow the rules of Kazakh grammar using natural language processing techniques.

REFERENCES

[1] "Modeling characteristics of agglutinative languages with multi-class language model for asr system" I. Dawa, Y.Sagisaka, and S.Nakamura, 2009.

[2] "Phonological foundations of the transition Kazakh alphabet to Latin graphics" Zeynep Muslimovna Bazarbayeva, Alimkhan Zhunisbek, Myrzabergen Malbakov, A.Baitursynov Institute of Linguistics, Ministry of Education and Science of the Republic of Kazakhstan, 2014.

[3] "Stem-based pos tagging for agglutinative languages" N. B ̈ol ̈uc ̈u and B. Can, 2017.

[4] "Болашаққа бағдар: рухани жаңғыру." N. Nazarbayev. Kazakhstan Institute of Strategic Studies under the President of the Republic of Kazakhstan, Astana, 2017.

[5] "Modelling of critical letter transmissions in Kazakh alphabet according to big data analysis" A. Issabek, R. Suliyev, G. Kessikbayeva, N. Sultanova, A. Bogdanchikov, K. Orynbekova, VESTNIK KazNRTU,ISSN 1680-9211, Almaty, 2019.

[6] "Distributed sentiment analysis of an agglutinative language via Spark by applying machine learning methods" Azamat Serek, Ainur Issabek, Andrey Bogdanchikov, ICECCO 2019, Nigeria.

[7] https://spark.apache.org/

[8] "Computing feature vectors of students for face recognition using Apache Spark", Darmen Kariboz, Andrey Bogdanchikov, Kamila Orynbekova, ICECCO 2019, Nigeria.

[9] "Open Information Extraction as Additional Source for Kazakh Ontology Generation", N Khairova, S Petrasova, O Mamyrbayev, 12th Asian Conference, ACIIDS 2020 Phuket, Thailand.

[10] "KazNLP: A Pipeline for Automated Processing of Texts Written in Kazakh", Zhandos Yessenbayev, Zhanibek Kozhirbayev, Aibek Makazhanov, Language22nd International Conference, SPECOM 2020 St. Petersburg, Russia.

[11] "A Parallel Implementation of the Pandas Framework", Saba Hafeez Khan, University of Houston 2020.

[12] "Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark", Ilias Mavridis, Helen Karatza, Journal of Systems and Software 2017.

[13] "Memory or Time: Performance Evaluation for Iterative Operation on Hadoop and Spark", Lei Gu, Huan Li, 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing.

[14] "Large Scale Distributed Data Science using Apache Spark", James G. Shanahan, Laing Dai, Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015.

[15] "Educational data mining with Python and Apache spark: a hands-on tutorial", Lalitha Agnihotri, Shirin Mojarad, Nicholas Lewkow, Alfred Essa, Proceedings of the Sixth International Conference on Learning Analytics and Knowledge, 2016.

М. Мералиев, К. Орынбекова, А. Таласбек*, Д. Карибоз, А. Исабек
Университет Сулеймана Демиреля, Казахстан
*e-mail: assem.talasbek@sdu.edu.kz

ОПТИМИЗАЦИЯ СЕГМЕНТОВ ДАННЫХ И КОЛИЧЕСТВА ЯДЕР ДЛЯ ОПРЕДЕЛЕНИЯ ПОПУЛЯРНОСТИ КАЗАХСКИХ СЛОВ С ПОМОЩЬЮ APACHE SPARK

Аннотация. Казахский язык является сложным агглютинативным языком. В данной работе мы использовали Apache Spark для оптимизации алгоритма выявления самых часто используемых слов в трех популярных казахских произведений. Основная цель заключалась в том, чтобы найти лучшую скорость вычислений, оптимизируя количество сегментов данных для определенного количества ядер. Данные были поделены на несколько сегментов и вычисления были выполнены на кластере с разным количеством ядер. Результаты показали, что скорость вычисления прямо зависит от количества сегментов данных.

Ключевые слова: Apache Spark, RDD, расчленение данных, NLP, MapReduce парадигма.

М. Мералиев, К. Орынбекова, А. Таласбек*, Д. Карибоз, А. Исабек
Сулейман Демирель Университеті, Қазақстан
*e-mail: assem.talasbek@sdu.edu.kz

ҚАЗАҚ СӨЗДЕРІНІҢ ТАНЫМАЛДЫҒЫН АНЫҚТАУ ҮШІН APACHE SPARK ПАЙДАЛАНУ АРҚЫЛЫ МӘЛІМЕТ СЕГМЕНТТЕРІН ЖӘНЕ ЯДРОЛАРДЫН САНЫН ОПТИМИЗАЦИЯЛАУ

Аңдатпа. Қазақ тілі құрылымы күрделі агглютинативті тіл. Берілген жұмыста біз Apache Spark-ті қолданыстағы үш танымал әдеби шығармаларында сөздердің жиілігін анықтау үшін қолдандық. Басты мақсат есептеудің ең жақсы жылдамдығын табу үшін берілген ядролардың саны бойынша мәліметтер сегменттерінің оңтайлы санын табу болды. Ол үшін мәліметтер бірнеше сегменттерге бөлінді және әр уақытта ядролардың саны әртүрлі кластерде өңделді. Нәтиже мәліметтер сегменттерінің саны есептеу жылдамдығына тікелей әсер ететіндігін көрсетті.

Негізгі сөздер: Apache Spark, RDD, Деректер бөлшектері, NLP, MapReduce парадигмасы.