UDC 629.3.025.2(045)
DOI:10.18372/1990-5548.78.18255

[1]D. O. Vedmiediev,
[2]N. V. Shapoval

TEXT MESSAGE CLUSTERING

National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine
E-mails: [1]vedmedevdanil@gmail.com, [2]shovgun@gmail.com ORCID 0000-0002-8509-6886

Abstract*—The division into groups of text messages is considered, which can be useful when building a personalized approach in different systems. To solve this problem, the Embedded Word2Vec was proposed. To enhance the division into groups, the suggestion of employing mini-batch k-means is presented, offering a method with lower computational demands. This recommendation aligns with the practical need for efficient and scalable clustering methods, especially when dealing with large datasets. Furthermore, the proposed metric based on the greatest common sequence is highlighted as a valuable tool for evaluating the similarity of texts. This metric not only serves as a means to assess clustering quality but also underscores the methodological approach of directly working with text data. The combination of these techniques presents a comprehensive framework for robust and effective text clustering, with potential applications in diverse fields, such as personalized system interactions and information retrieval.*

Index Terms—Text message analysis; machine learning; Embedded Word2Vec; Mini Batch K-means; longest common subsequence method; clustering; SMS.

I. INTRODUCTION

A contemporary individual consistently engages in communication through email, short text messages on social networks, or text messengers. The continual influx of a substantial amount of textual information necessitates thorough analysis to address numerous challenges. Consequently, addressing the issue of identifying key topics enables the effective structuring and organization of extensive textual information, thereby facilitating subsequent analysis. The examination of user sentiment in text messages can be leveraged to forecast trends and respond to shifts in public opinion. The amalgamation of text messages into clusters permits a personalized approach, enhancing user experience across various systems. By categorizing messages based on similarities, users can receive more targeted and captivating content tailored to their interests and preferences. This aspect holds particular significance in the realm of marketing and advertising, where precision-targeted communication can significantly enhance the efficacy of campaigns.

An important consideration lies in the capability to process information in real-time, a task that poses challenges when dealing with substantial data volumes. Swift and precise analysis of text messages enables prompt responses to inquiries, complaints, or other crucial facets of communication—a critical necessity in various domains such as business, society, and resource management. The present study is centered on the development of a tool designed to detect patterns and group semantically similar messages.

II. PROBLEM STATEMENT

A. Methods for text message analysis

Contemporary research extensively employs machine learning methods for text analysis, incorporating word embedding representations and K-means clustering. Nonetheless, the majority of these approaches are constrained to the analysis of extensive text corpora rather than specific formats, such as text messages.

For example, in [1], large texts were represented in matrix form, and then, using Power Iteration Clustering, singular vectors of similarity were derived from these matrices. A drawback of this approach is that the transition to the matrix space results in a clustering that is notably large-scale and computationally expensive. In the works of [2] and [3], spectral clustering of matrices was conducted utilizing methods that identify a low-dimensional embedding linked to the eigenvectors of the similarity matrix, followed by the application of the K-Means method to obtain the final clusters. The advantage of these approaches lies in the fact that transitioning to clustering requires only the corresponding feature vectors of each text that is to be clustered.

In the study [4], the task of clustering short text responses using Embedded Word2Vec was partially

solved. Feature vectors of a fixed size corresponding to each text were obtained, and the transition to clustering was executed through the application of the K-Means method, leading to the formation of the final clusters for the texts.

Embedded Word2Vec represents a word vectorization method that uses neural networks to acquire numerical representations of words as vectors with a fixed length. It is based on the Word2Vec algorithms, which employ deep neural networks to explore the semantic relationships among words in the text. This approach makes it easier to analyze texts, find similar words, and solve natural language processing problems.

In works [5] and [6], a study was carried out to optimize the solution to the problem of finding a subsequence among a set of lines, which has two properties: it is common to all and is the longest. The Longest Common Subsequence method can be used to output the longest common subsequence between given sets of strings. Two important conclusions can be drawn from this: firstly, the text can be represented as a sequence of both words and symbols; secondly, it is possible to derive a numerical metric that will describe each instance of the text as the number of common characters between the instances of the cluster and the center of the cluster.

In work [7], a comparison between the K-means and mini-batch K-means methods was conducted through an analysis of their performance metrics. The findings led to the conclusion that mini-batch K-Means represents a superior and more contemporary approach compared to traditional K-means. It addresses some of its drawbacks, including reduced memory usage, compatibility with large datasets in memory, and shorter convergence time. This improvement is attributed to the batch clustering of points rather than processing the entire dataset.

In summary, the utilization of mini-batch K-Means is advantageous for handling large datasets, and the generation of feature vectors using Embedded Word2Vec offers a promising solution to the clustering of short SMS messages.

This study focuses on the analysis of a large set of data, which includes SMS messages of various nature and related performance problems and ways of presenting this data for analysis.

B. Evaluation of clustering quality

Classical metrics commonly employed to assess the quality of clustering include the silhouette coefficient, inertia, Calinski–Harabasz index, and Davies–Bouldin index. The silhouette coefficient gauges the similarity of each object within a cluster to other objects in the same cluster, relative to objects in other clusters. Inertia represents the sum of squared distances of samples to the nearest cluster center, weighted by the sample's specified weight.

The Calinski–Harabasz metric is computed based on intra- and inter-cluster variances. On the other hand, the Davies–Bouldin index assesses the "internal compactness" of clusters and the "separation" between them. This metric compares the average intracluster distance between each point in a cluster and its center with the average intercluster distance to the nearest cluster. However, it is important to note that these metrics may not accurately reflect the quality of clustering for textual data.

Based on the above, the following tasks must be addressed to build clusters of text messages:

1. Processing a large array of text messages quickly and efficiently.
2. Evaluate the quality of the proposed clustering based on textual information.

III. PROBLEM SOLUTION

A. SMS messages clustering algorithm

To solve the problem of clustering short text messages, such as SMS, the following algorithm are proposed:

1) use the embedded vector representation of words (Embedded Word2Vec);

2) use Mini Batch K-means for further clustering;

3) use the method of the largest common subsequence for clustering analysis;

4) based on the analysis of the results, repeat steps 1–3.

This approach suits for efficient detection of patterns and grouping of similar messages in large data sets, which is important for analyzing communication flows and identifying communication features.

For evaluation of clustering quality, it is proposed to use the Longest Common Subsequence (LCS) method. LCS is one of the methods for evaluating the similarity between texts. It is used to determine the degree of similarity between two texts by finding the largest sequence of characters that is stored in the same order in both texts. It differs from the considered metrics in that it works not with vectors, but directly with texts obtained as a result of clustering.

Initially, the approach involves the consideration of a cluster and its center, followed by the

determination of the longest common subsequence between them. This length is quantified by the number of characters shared between both texts. Subsequently, the length of this common subsequence undergoes normalization by dividing it by the total length of the text instance (or information within the cluster). This resulting value provides a percentage indicating the similarity between the cluster and its center. Consequently, the LCS method has been proposed as a metric for evaluating the similarity of two texts.

By employing this metric, the "purity" of clusters can be accessed through measures such as the mean and lower quartile of each sample within a cluster. Alternatively, it can be used to establish limits on the degree of confidence for each element within the cluster. For instance, a condition could be set, such as: "if an instance of a given cluster exhibits a similarity to its center of p%, then this instance belongs to this cluster," where $p = \{60, 85, 90\}$.

For example, if we have the texts "cluster" and "cluster center", then the largest common subsequence will be "cluster", and after normalizing the ratio of lengths, we will get a high similarity index. This approach allows for the assessment of the clustering results with precision in terms of what is signified by the expression "similarity of two texts".

For text embedding and clustering, it is suggested to use Embedded Word2Vec and the mini-batch K-Means, which demonstrate excellent performance.

B. Experimental results

For experiments, 20 templates (clusters) were generated, totaling 500,000 texts. The results presented in Table I specify the effectiveness of clustering for ten and twenty clusters.

Thus, the closer the Silhouette coefficient is to 1, the more effective the clustering, indicating a high degree of similarity among objects within clusters. Conversely, a Silhouette coefficient closer to 0 may suggest the presence of diverse objects within a single cluster. Minimizing the Inertia metrics and Davies–Bouldin coefficient contributes to improved clustering. Simultaneously, maximizing the Calinski–Harabasz coefficient enhances the quality of clustering.

An experiment was conducted in which the number of clusters was intentionally set to $N = 10$, despite the presence of 20 templates. Subsequently, 10 clusters were obtained through the clustering process. Following this, it became necessary to identify the SMS closest to the center in each cluster, referred to as "cluster centers."

TABLE I. COMPARISON OF CLUSTERING METRICS

Number of clusters / Metrics	***N* = 10**	***N* = 20**
Silhouette coefficient	0.6	1
Inertia	1167051.7	7.46E-26
Calinski–Harabasz coefficient	258068.8	715313510659.4
Davies–Bouldin coefficient	0.78	0.0005

Upon obtaining the "cluster centers," the proposed similarity metric was applied to each cluster. This approach facilitated the selection of a considerable number of correctly clustered results for each cluster, representing a partial outcome of the clustering process. Furthermore, the threshold of similarity can be flexibly chosen, thanks to the algorithm's intuitive application. Figure 1 illustrates the distributions of the LCS metric for each cluster, revealing instances where the clustering did not perform optimally, specifically for the predetermined number of clusters $N = 10$ (see Table I).

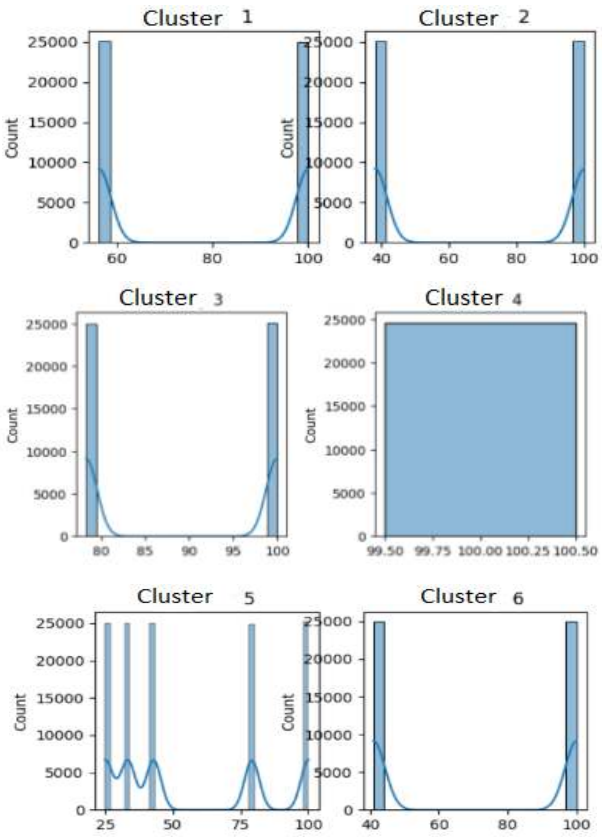

Fig. 1. Distribution of LCS metrics for each cluster

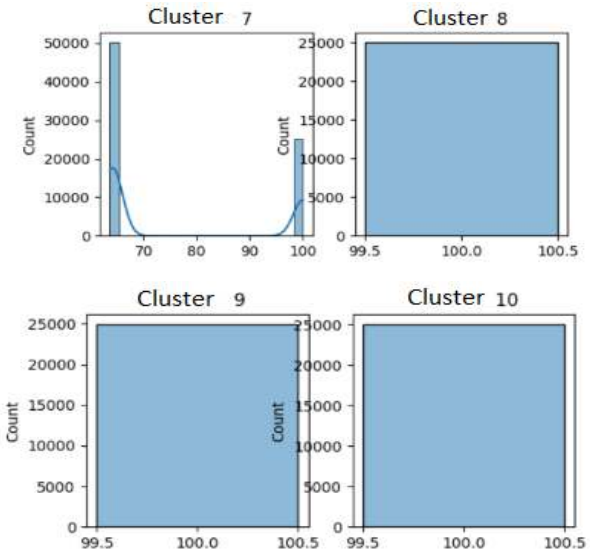

Fig. 1. Ending. (See also p. 18)

In Figure 1, it is evident that clusters 4, 8, 9, and 10 were effectively declustered and appear "clean." However, all other instances within the cluster must undergo a second round of clustering (re-processing).

If, despite these efforts, the results do not entirely meet the task requirements (e.g., not all clusters are identified, but only a portion), the procedure can be iterated for the data that did not fall into the clusters of the partial result. Subsequently, employing the same metric, all similar centers of each cluster obtained from the repeated procedure are aggregated into a unified cluster, thereby concluding the entire clustering process.

IV. Conclusion

To cluster text messages effectively, the utilization of a vector representation of words is imperative. Additionally, for the evaluation of text clustering, a metric that operates directly with the text itself is indispensable. The proposed metric, founded on the greatest common subsequence method, fulfills this requirement, enabling precise assessment of clustering quality.

Expanding upon these findings, it is noteworthy that the proposed algorithm, as demonstrated through the conducted experiments, exhibits not only high accuracy but also remarkable stability in partitioning SMS messages into pertinent clusters. Furthermore, the robustness of the algorithm underscores its potential applicability in diverse contexts, emphasizing its reliability and efficacy in the clustering of short text messages.

References

[1] Frank Lin and William W. Cohen, “A Very Fast Method for Clustering Big Text Datasets,” *In: Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, 2010, pp. 303–308.

[2] Andrew Ng, Michael Jordan, and Yair Weiss. “On spectral clustering: Analysis and an algorithm,” *Advances in neural information processing systems*, 2001, 14.

[3] Von Luxburg, Ulrike. *A Tutorial on Spectral Clustering. Statistics and Computing. Data Structures and Algorithms* (cs. DS); Machine Learning, pp. 395–416.

[4] Rohan Saha, ‘Influence of various text embeddings on clustering performance in NLP’, 2023.

[5] Abdi A., Hajsaeedi M., Hooshmand M., "Longest Common Substring in Longest Common Subsequence's Solution Service: A Novel Hyper-heuristic," *Computational Biology and Chemistry*, vol. 105, p. 107882, 2023. https://doi.org/10.1016/j.compbiolchem.2023.107882

[6] Negev Shekel Nosatzki, “Approximating the Longest Common Subsequence problem within a sub-polynomial factor in linear time,” arXiv e-prints, 2021, https://doi.org/10.48550/arXiv.2112.08454

[7] G. Yamini, Dr. B. Renuka Devi, “A New Hybrid Clustering Technique Based on Mini-batch K-means and K-means++ for Analysing Big Data,” *International Journal of Recent Research Aspects*, 2018.

[8] Carl Allen and Timothy Hospedales, “Analogies Explained: Towards Understanding Word Embeddings,” *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97:223–231, 2019.

Received September 04, 2023

Vedmiediev Daniil. Master's degree student.
Department of Mathematical Methods of System Analysis. Institute of Applied Systems Analysis, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.
Research area: Unsupervised machine learning, Supervised machine learning, Deep learning.
E-mail: vedmedevdanil@gmail.com

Shapoval Nataliia. ORCID 0000-0002-8509-6886. Candidate of Science (Engineering). Associate Professor.
National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine.
Education: Kyiv Polytechnic Institute, Kyiv, Ukraine, (2010).
Research interests: computer vision, fuzzy neural network, deep neural network.
Publications: 8.
E-mail: shovgun@gmail.com

Д. О. Ведмедєв, Н. В. Шаповал. Кластеризація текстових повідомлень

Розглянуто поділ текстових повідомлень на групи, що може бути корисним при побудові персоналізованого підходу в різних системах. Для вирішення цієї проблеми був запропонований вбудований Word2Vec. Пропонується використання mini-batch k-means, як методу із меншими обчислювальними вимогами. Ця рекомендація узгоджується з практичною потребою в ефективних і масштабованих методах кластеризації, особливо при роботі з великими наборами даних. Крім того, запропонована метрика, заснована на найбільшій загальній послідовності, виділяється як цінний інструмент для оцінки подібності текстів. Цей показник не тільки служить засобом оцінки якості кластеризації, але й підкреслює методологічний підхід безпосередньої роботи з текстовими даними. Поєднання цих методів представляє комплексну структуру для надійної та ефективної текстової кластеризації з потенційними застосуваннями в різноманітних сферах, таких як персоналізована взаємодія системи та пошук інформації.

Ключові слова: аналіз текстових повідомлень; машинне навчання; Embedded Word2Vec; Mini Batch K-means; метод найбільшої спільної підпослідовності; кластеризація; СМС-повідомлення.

Ведмєдєв Данило Олексійович. Студент магістр.
Кафедра математичних методів системного аналізу, Інститут прикладного системного аналізу, Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, Україна.
Напрям наукової діяльності: Машинне навчання, аналіз текстових даних, глибокі нейронні мережі.
E-mail: vedmedevdanil@gmail.com

Шаповал Наталія Віталіївна. ORCID 0000-0002-8509-6886. Кандидат технічних наук. Доцент.
Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, Україна.
Освіта: Київський політехнічний інститут, Київ, Україна, (2010).
Напрям наукової діяльності: комп'ютерний зір, нечіткі нейронні мережі, глибокі нейронні мережі.
Кількість публікацій: 8.
E-mail: shovgun@gmail.com