The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 1376 - 1385

Generation of Regular Expression from Aligned Sequences of Text Snippets

Dr. Girishkumar K. Patnaik
Professor, Department of Computer Engineering, SSBT, COET, Jalgaon,
patnaik.girish@gmail.com

Mr. Dinesh D. Puri
Research Scholar, Department of Computer Engineering, SSBT, COET, Jalgaon,
ddpuri@gmail.com

Mr. Akash D. Waghmare
Research Scholar, Department of Computer Engineering, SSBT, COET, Jalgaon,
aakashjan8@gmail.com

Abstract

Introduction: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Vitae sapien pellentesque habitant morbi tristique senectus et netus. Dignissim cras tincidunt lobortis feugiat vivamus at augue eget arcu. At risus viverra adipiscing at in. Cras semper auctor neque vitae tempus quam. Sed cras ornare arcu dui. Turpis massa sed elementum tempus. Risus commodo viverra maecenas accumsan lacus vel facilisis volutpat est.

Objectives: Effective data management has become an absolute necessity in today's world as a direct result of the widespread adoption of electronic medical records for patients. For medicine and healthcare domains, particularly effective web healthcare providers, computer-assisted categorization of such information into functional classifications such as medical issues or illnesses may save time and efforts. However, a substantial volume of data in the healthcare domain is still unstructured.

Methods: The utilization of unstructured text datasets is a challenge in comparison with structured text datasets. Many researchers in various domains have proposed to convert unstructured text into structured text. The text needs to be built in a fixed and aligned pattern for easy comparison, matching, and classification. In the text classification process, accuracy is one of the challenges when text is unstructured. The text with a fixed pattern is easy to classify. The regular expressions are sequences of characters with fixed patterns of text. This strength of regular expression is used in the proposed text classification. Finding patterns in unstructured text is possible with the use of sequence alignment. Basically, sequence alignment is a concept of biomedical research, but the same concept is used to align text snippets that are not perfectly similar to each other.

Results: The sequence alignment technique detects the similarity score assigned to each potential alignment in order to choose among the numerous local alignments of sequences. The proposed local pairwise alignment method is used to get sequence alignments, which are useful to generate regular expressions.

Conclusions: A regular expression is a string of characters used to describe a text pattern. The regular expressions are typically created manually by the domain experts. The proposed work is the automatic generation of regular expressions from aligned sequences with a bottom up approach. The generated regular expressions are used as a dataset on which various machine learning algorithms are applied for text classification.

Keywords: Sequence Alignment, Regular Expression, Classification, Machine learning, NLP

1. Introduction

The process of sequence alignment is an essential component where sequence matching is necessary, as in the drug design process, it enables the identification of aberrant alterations in the protein sequences that the drug is exposed to. As a direct consequence of this, it is possible to develop an effective medicine with fewer adverse effects [1, 2, 3, 4]. As a result, sequence alignment is currently playing an important and significant part in the progression of human existence and society. The majority of the research that has been done up to now has

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 1376 - 1385

concentrated on finding the most efficient way to implement and parallelize the numerous relevant sequence alignment techniques, such as the Smith-Waterman algorithm. The analysis of these algorithms has received almost no attention.

The characteristics of sequence alignment are useful for processing the unstructured data. The regular expressions are having fix pattern of characters. If the text has fix and uniform structure, building of regular expression from it is quite easy. However, for unstructured text, formation of regular expression is challenging. The existing text classifiers use traditional approach along with text dataset for text classification. Utilization of regular expressions strength can accelerate the performance of regular expression based classifiers. The paper is organized as follows. Section II presents the literature review related to Sequence Alignment and Regular Expression Generation. Section III depicts the method of proposed system for regular expression generation. Section IV states the result and discussion. Finally, Section V concludes the paper and provides future direction.

2. Related Work

Chowdhury et al., in [5], described types of sequence alignment and methods applied in sequence alignment. The authors stated two alignment methods, local alignment and global alignment. Global alignment attempts to match as much of the sequence as possible. The tool for Global alignment is based on Needleman-Wunsch algorithm. Local alignment is used to find the regions with highest density of matches. The local alignment uses on Smith-Waterman algorithm.

Hogeweg et al., in [6], stated progressive alignment for multiple sequence alignment (MSA). Progressive is a heuristics approach where complex MSA problem is separated into sub problems. The same approach solves direct MSA problem indirectly with PSA. Heuristics approach assembles all sequences progressively where best pair wise alignment is first taken into account. In the same approach multiple sequences are matched at a time due to which possibility of error is increased. Progressive alignment uses guide tree to solve MSA problem where each leaf represents a sequence to be aligned.

Feng et al., in [7], introduced guide tree method in progressive alignment. The guide tree guides the merging order of sequences based on the pairwise distances calculated for all the possible sequence pairs to be aligned in MSA. However, the guide tree causes errors in progressive alignment if an error is introduced while measuring the distances or at the time of tree construction. Ultimately, the error is reflected in the final alignment. The problem can be solved by iterative methods by repetitively modifying the guide tree and calculating the distance measurement. Hence, iterative methods are better approaches.

Wang et al., in [8], introduced iterative approach to enhance progressive alignment. The iterative method generally performs post-processing by making changes in the alignment made by progressive methods. The iterative approach modifies the construction of the guide tree.

Mahabhashyam et al., in [9], used popular statistical models like Hidden Markov model for sequence alignment. The authors used ProbCons method which uses a new scoring function based on probabilistic consistency. Hidden Markov model is also a progressive approach that uses a combination of probabilistic modeling and consistency-based alignment techniques.

Stoye et al., in [10], worked for the reduction of the search space of MSA. The authors used the principle of the Divide and Conquer algorithm to divide and concatenate the MSA. The mentioned approach first identifies the "optimal cut" points using pairwise projected alignments for partitioning a large multiple alignment into smaller subproblems. Each of the small parts is aligned separately and then joined to produce the final MSA.

Lassmann et al., in [11], stated a scoring model for sequence alignment. The sequence alignment techniques quantitatively measure the quality of an alignment by considering a scoring model. The most commonly used scoring model chosen by several MSA methods is the sum of pairs (SOP). The scoring model is an extension of

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 1376 - 1385

the typical pairwise scoring technique to a multiple alignment approach. Here, a pairwise matched residue gets a positive score, a mismatch gets a negative score, and a space or gap gets a negative score.

Wang et al. in [12], addressed the disadvantage of the sum of pairs method, which requires exponentially more computational time and memory as the number of sequences increases. For MSA, all the possible sequence pairs are scored first based on pairwise scoring, and after that, all the paired scores are summed to get the total SOP score. The SOP score is defined where δ (Si, Sj) is the pairwise alignment score between the two aligned sequences Si and Sj; k is the total number of sequences.

Menglin et al., in [13], developed a unique constructive heuristic technique, for medical text classification tasks that generates regular expression based classifiers. The method just requires a set of labeled samples and has no restrictions on the alphabet's size. The authors tested the system on actual medical data and found that it performed well and was consistent. The machine-generated regular expressions could be employed efficiently in combination with machine learning approaches to conduct medical text categorization tasks, according to the experimental outcomes.

Chaofan et al., in [14], proposed PSAW algorithm, which is a regular expression learning algorithm that combines pool-based simulated annealing as well as a word vector model to meet the criteria of interpretability and readability in the healthcare profession field. PSAW outperformed domain experts on 30 clinical text classification tasks, each involving half a million actual records from one of China's major online healthcare platforms. Most of PSAW's classifiers are human readable for additional changes and validation.

3. Objectives

The emphasis of the work proposed in this paper is on regular expression based English text classification. Natural language processing applications typically use regular expressions that have been developed manually by human experts. The goal is to automate both the creation and utilization of regular expressions in text classification. The problem deals with the generation of regular expressions from aligned sequences.

The paper proposes a regular expression generation method to generate regular expressions, which are further used in text classification. The proposed method generates regular expressions with bottom up approach. The group of aligned tokens produces as keys which help to build patterns of regular expression from aligned sequences. The filteration process discards low quality regular expressions based on threshold precision

4. Methods

The sentence, text, or any unstructured data is considered as a text snippet. A text snippet is defined as a group of printable characters that offers semantic context for the text's categorization. The text snippets are input to the generation of the regular expression module. The text snippets are aligned using proposed pairwise sequence alignment method with appropriate alignment parameters such as match, mismatch, and gap value. The generation of regular expressions is explained where four text snippets S1, S2, S3 and S4 are considered as examples.

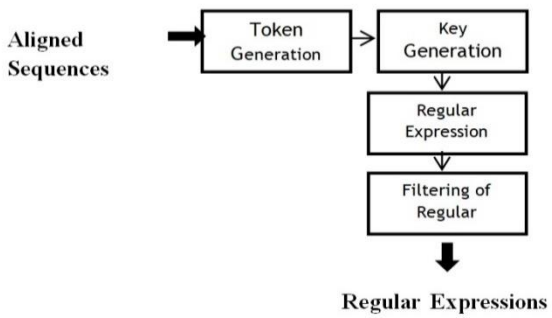


Figure 1. Regular Expression Generation

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 1376 - 1385

S1: he consumes 1 pack of tablets a day
S2: he consumes five tablets per day
S3: patient consumes one pack a day
S4: he continues to consume one-half pack per day

Token Generation:
The text snippets are aligned with the proposed sequence alignment method. The tokens are generated from aligned sequences. Tokens, which are classified as words, numbers, or symbols, are found in snippets. With the help of the Stanford NLP group's Penn Treebank tokenizer, the text snippets are first converted to lowercase and tokenized.

Key Generation:
The group of tokens is considered as phrase. A key is a list of phrases in a particular order. The key divides distinct phrases with one or more tokens. The Figure 4.2 shows the key generation method from aligned sequences. For example, the sequences S1 and S2 are aligned. The aligned tokens from both sequences are [he consumes, day]. The same keys are used to build the patterns for regular expressions.

S1: *he consumes* 1 pack of tablets a *day*

S2: *he consumes* five tablets per *day*

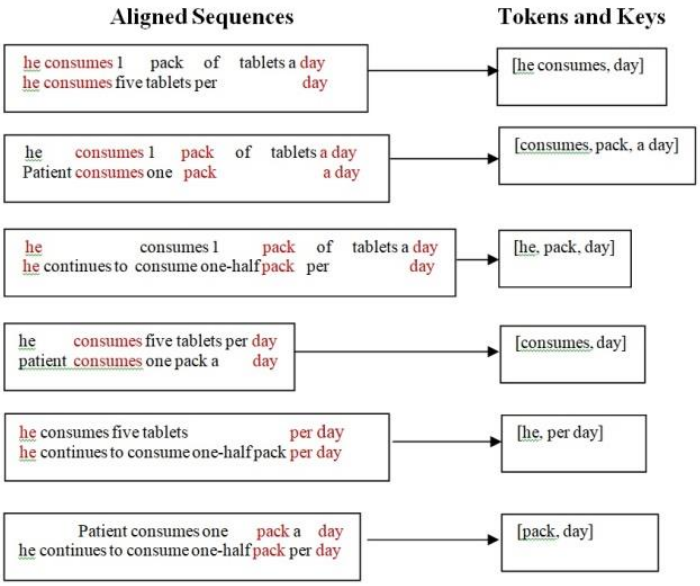


Figure 2. Key Generation from Aligned Text

Regular Expression Generation:
For each key, the proposed method produces two regular expressions, with and without distance control. The (\s+\S+)* pattern allows for any number of tokens to be placed between phrases in the non-distance control approach, which develops regular expression using phrases from keys. The numbers of tokens between phrases are constrained by the distance control mechanism. It employs the pattern (\s+\S+){n,m}, where \S+ denotes any collection of non-whitespace letters that together form a word, "\s" denotes a whitespace character, n denotes the minimum token length and m denotes the maximum token length permitted. The Table 1 shows patterns for regular expression.

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 1376 - 1385

Table 1. Generated Regular Expression

Key: [he consumes, day] Regular Expressions: he\s+consumes(\s+\S){4,5}\s+day he\s+consumes(\s+\S)*\s+day	Key: [consumes, pack, a day] Regular Expressions: (\s+\S){1}consumes(\s+\S){1}\s+packs(\s+\S){0,2}\s+a\s+day consumes(\s+\S)*\s+packs(\s+\S)*\s+a\s+day
Key: [he, per day] Regular Expressions: he (\s+\S){3,5}\s+per \s+day he (\s+\S)*\s+per \s+day	Key: [pack, day] Regular Expressions: (\s+\S){3,5} pack(\s+\S){1,3}\s+day pack(\s+\S)*\s+day
Key: [he] Regular Expressions: he(\s+\S+){5,7} he	Key: [consumes, day] Regular Expressions: (\s+\S){1}\s+consumes(\s+\S){3,5}\s+day consumes(\s+\S)*\s+day

Filteration of Regular Expression:

Each new regular expression generated with a key is evaluated against the rest of the regular expressions in the training set. For each distinct expression, the metrics for recall, precision, and F-measure are computed. Using precision as the threshold, the regular expressions are filtered. Since human developers often build expressions with 100% precision, but for proposed method to generate regular expression threshold precision is 90%. The majority of the regular expressions generated from keys are eliminated by the high precision threshold. The filtering of regular expressions is shown in Table 2.

Table 2. Filteration of Regular Expression

Key: [he consumes, day] Regular Expressions: he\s+consumes(\s+\S){4,5}\s+day he\s+consumes(\s+\S)*\s+day	Key: [consumes, pack, a day] Regular Expressions: (\s+\S){1}consumes(\s+\S){1}\s+packs(\s+\S){0,2}\s+a\s+day consumes(\s+\S)*\s+packs(\s+\S)*\s+a\s+day
Key: [he, per day] Discarded Regular Expressions: he (\s+\S){3,5}\s+per \s+day he (\s+\S)*\s+per \s+day	Key: [pack, day] Discarded Regular Expressions: (\s+\S){3,5} pack(\s+\S){1,3}\s+day pack(\s+\S)*\s+day
Key: [he] Discarded Regular Expressions: he(\s+\S+){5,7} he	Key: [consumes, day] Regular Expressions: (\s+\S){1}\s+consumes(\s+\S){3,5}\s+day consumes(\s+\S)*\s+day

After filteration process, remaining algorithms are used for classification. The Table 4.3 shows filtered regular expression.

Table 4.3 Filtered Regular Expression

he\s+consumes(\s+\S){4,5}\s+day
he\s+consumes(\s+\S)*\s+day
(\s+\S){1}consumes(\s+\S){1}\s+packs(\s+\S){0,2}\s+a\s+day
consumes(\s+\S)*\s+packs(\s+\S)*\s+a\s+day
(\s+\S){1}\s+consumes(\s+\S){3,5}\s+day
consumes(\s+\S)*\s+day

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 1376 - 1385

5. Learning of Generated Regular Expression

The Naïve Bayes classifier is used for learning regular expressions. The classifier works for learning of filtered regular expressions and unfiltered regular expressions. The training Algorithm 1 is used for module training and Algorithm 4.3 used for testing. Once regular expression generation has been done, the data is split into 75-25% for training and testing, respectively. After reading the data, the Porter-Stemming algorithm [36] is used to extract stop word and lemmas features. All extracted features are considered when building a classifier.

Algorithm 1: Naïve Bayes Training Algorithm

Input: Training dataset TrainData[],
Various activation functions[], Threshold Th
Output: Extracted class wise Features Feature_set[]
for complete trained module.
1. Set input block of data d[], activation function, epoch size,
2. Features.pkl←ExtractFeatures(d[])
3. Feature_set[] ← optimized(Features.pkl)
4. Return Feature_set[]

The Algorithm extracts features such as lemmas from input text. For data filtration, stop words are removed, as are null values. When the test classifier generates a similarity score between two input objects, both are required as input. These are two separate attributes that represent the training and testing instances, respectively. The Th is the denominator that used for selection of each epoch layer result. The A[j] denotes jth attributes of testing instance while the A[k] depicts kth train attribute information. By using feature selection method, the features are extracted from both instances and forward to the similarity measurement function.

Algorithm 2: Naive Bayes Testing Algorithm

Input: Testing dataset TestDBLits [], Train dataset TrainDBLits[] and Threshold Th.
Output: Resultset<class_name, Similarity_Weight> all set which weight is greater than Th.
1. For each testing records as given below equation

$$\text{testFeature(k)} = \sum_{m=1}^{n} (.\,\text{featureSet[A[i] … …. . A[n]} \leftarrow \text{TestDBLits})$$

2. Create a feature vector from testFeature(m) using the below function.
   Extracted_FeatureSet_x [t…..…n] = $\sum_{x=1}^{n}(t)$←testFeature (k)
   Extracted_FeatureSet_x[t] holds the extracted feature of each instance for the testing dataset.
3. For each train instances as using the below function

$$\text{trainFeature(l)} = \sum_{m=1}^{n} (.\,\text{featureSet[A[i] … …. . A[n]} \leftarrow \text{TrainDBList})$$

4. Generate new feature vector from trainFeature(m) using below function
   Extracted_FeatureSet_Y[t……n] = $\sum_{x=1}^{n}(t)$←TrainFeature (l)
   Extracted_FeatureSet_Y[t] holds the extracted feature of each instance for the training dataset.
5. Evaluate each test records with the entire training dataset

$$\text{weight} = \text{calcSim (FeatureSetx} \,||\, \sum_{i=1}^{n} \text{FeatureSety[y])}$$

6. Return Weight
7. End

6. Result

The Drug review dataset is taken from www.kaggle.com which contains six attributes and 215063 records. The dataset contains large text data with user comment and disease associated with it. In the Table 4 the information of dataset is given.

Table 4 Information of Dataset

Dataset	Class Attributes	No. of instances
Training data	Depression	600
	Birth Control	550
	Pain	940

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 1376 - 1385

	Bipolar Disorder	1150
	Weight Loss	890
Testing dataset	Depression	240
	Birth Control	205
	Pain	380
	Bipolar Disorder	450
	Weight Loss	385

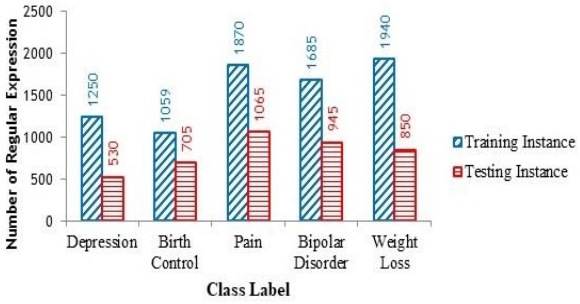
The Table 5 describes attribute information such as comment_id, comment and class associate with comment. The comment is unstructured text which is aligned using a sequence alignment algorithm.

Table 5. Attribute Description of Dataset

	Attribute name	Attribute type
1	comment_id	Numeric
2	comment	String
3	Class	string

In addition to similar circumstances, the dataset includes patient feedback on individual medications, which differ dramatically in patient rating, indicating patient satisfaction and quality. Clambering online pharmacy reviews have collected the details. Drug impression sentiment classification over various facets, i.e. attitudes acquired on particular factors such as efficacy and side effects, the generalizability of models between domains, i.e. circumstances, and domains, i.e. model interpretation from multiple data.

The dataset is partitioned into a test (25 %), train (75%) and contained within two .csv directories. For the execution of the program IDE (Netbeans 8.0) is used which implemented in Java (JDK 1.8). The Figure 2 shows the number of regular expression generated from dataset using the proposed regular expression generation method. The learning of regular expression is based on quality of generated regular expression. The quality of regular expression is maintained through filtering process. The filtering process helps to survive those regular expressions which are having higher learning capacity.



The Figure 3 shows the comparative analysis of performance parameters for generated regular expressions.

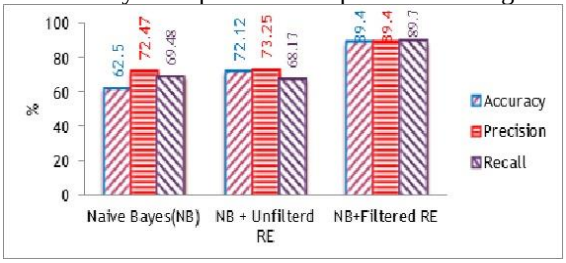


Figure 3. Performance Evaluation of Proposed System with Existing Algorithm

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 1376 - 1385

7. Discussion

To analyze the quality of generated regular expressions three systems are implemented. The traditional classifiers Naïve Bayes, Naïve Bayes with unfiltered regular expressions, and Naïve Bayes + filtered regular expression are used. The traditional Naive Bayes classifier is used for the classification of the text snippet dataset. In the second approach, unfiltered regular expressions generated from aligned sequences using a proposed sequence alignment algorithm are used for classification. The gap parameter value used in this approach is -2. Finally, filtered regular expressions generated from aligned sequences using the proposed sequence alignment algorithm with a gap parameter -1 are used. In the filteration process those regular expression which fail to achieve threshold precision are discarded. The use of unfiltered regular expressions fails to correctly classify the text. The gap value variation produces more accurate aligned sequences, which are used to create accurate regular expressions. Numbers of iterations are performed to generate correct regular expressions so the accuracy of regular expressions is improved. Experimental results show that the accuracy of the regular expression based Naïve Bayes classifier up to 90% which is higher than existing approaches. If the input is filtered regular expression, it impacts on accuracy of classification as compared to unfiltered regular expressions. The results show that regular expression based classifier is more effective than traditional classifiers. The difference between accuracy of traditional Naïve Bayes classifier and regular expression based classifier is around 27% which proves that regular expression based classifiers produce more accuracy than traditional classifier.

8. Conclusion

The generation of regular expressions from unstructured text was a challenging and time-consuming activity due to the complexity of the patterns. But due to aligned sequences, it became easy to generate uniform patterns, which are further used to generate regular expressions. The proposed method generates regular expressions from aligned string sequences where keys are generated, and from keys, regular expressions are generated. For improvement in performance, the filtering of regular expressions is performed based on the threshold precision value, which helped a lot to generate accurate regular expressions. In the proposed work, experimental investigation is conducted to enable the accurate generation of regular expressions.

References

[1] Chaofan Tu, Ruibin Bai, Zheng Lu, Uwe Aickelin, Peiming Ge and Jianshuang Zhao. “Learning Regular Expressions for Interpretable Medical Text Classification”, IEEE Access. pp. 1-4, 2019, doi:10.1109/cec48606.2020.9185650

[2] Musarrat Hussain, Jamil Hussain, Taqdir Ali and Sungyoung Lee. “An Empirical Method of Automatic Pattern Extraction for Clinical Text Classification”, 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, pp. 5292-5295, 2020, DOI:10.1109/EMBC44109.2020.9176503

[3] Disheng Pan, Xizi Zheng, Weijie Liu, Mengya Li, Meng Ma, Ying Zhou, Li Yang and Ping Wang. “Multi-label Classification for Clinical Text with Feature-level Attention”, 6th Intl Conference on Big Data Security on Cloud, IEEE. pp.186-195, 2020 doi:10.1109/bigdatasecurity-hpsc-ids49724.2020.00042

[4] Mehraj, H., Jayadevappa, D., Haleem, S. L. A., Parveen, R., Madduri, A., Ayyagari, M. R., & Dhabliya, D. (2021). Protection motivation theory using multi-factor authentication for providing security over social networking sites. Pattern Recognition Letters, 152, 218-224. doi:10.1016/j.patrec.2021.10.002

[5] Flores, C.A., Figueroa, R.L., & Pezoa, J.E. FREGEX: A Feature Extraction Method for Biomedical Text Classification using Regular Expressions. 2019 41st Annual

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 1376 - 1385

International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 6085-6088 , 2019, DOI: 10.1109/EMBC.2019.8857471

[6] Biswanath Chowdhury, Gautam Garai. "A Review on Multiple Sequence Alignment From the Perspective of Genetic Algorithm" Journal of Genomics, Volume 109, Issues 5, pp. 419-431, 2017, https://doi.org/10.1016/j.ygeno.2017.06.007.

[7] P. Hogeweg, B. Hesper, "The Alignment Of Sets Of Sequences and the Construction of Phylogenetic Trees: An Integrated Method", Journal of Molecular Evolution, Volume 20, Issue 2, pp. 175–186, 1984, https://doi.org/10.1007/BF02257378

[8] Raghavendra, S., Dhabliya, D., Mondal, D., Omarov, B., Sankaran, K. S., Dhablia, A., . . . Shabaz, M. (2022). Development of intrusion detection system using machine learning for the analytics of internet of things enabled enterprises. IET Communications, doi:10.1049/cmu2.12530

[9] D.F. Feng, R.F. Doolittle, "Progressive Sequence Alignment as a Prerequisite To Correct Phylogenetic Trees", Journal of Molecular Evolution (J Mol Evol)-Vol. 25, Iss: 4, pp 351-360, 2005, doi: 10.1007/BF02603120

[10] Y. Wang, K.B. Li, "An Adaptive and Iterative Algorithm for Refining Multiple Sequence Alignment", Journal: Computational Biology And Chemistry. Vol. 21, pp 3615–3621, 2005, doi:10.1093/bioinformatics/bti582

[11] C. B. Do, M. S. Mahabhashyam, M. Burdon, S. Batzoglou, ProbCons : "Probabilistic Consistency-Based Multiple Sequence Alignment", Journal: Genome Research. pp 330–340. 2005, doi: 10.1101/gr.2821705

[12] Jens Stoye, Vincent Moulton, Andreas W.M. Dress, "DCA: An Efficient Implementation of the Divide-And-Conquer Approach to Simultaneous Multiple Sequence Alignment", Bioinformatics, Volume 13, Issue 6, Pages 625–626, 1997, https://doi.org/10.1093/bioinformatics/13.6.625

[13] T. Lassmann, E. Sonnhammer, "Quality Assessment of Multiple Alignment Programs", Fedration of Europian Biochemical scientist (FEBS) Lett. 529, pp. 126–130, 2005, doi: 10.1016/s0014-5793(02)03189-7

[14] L. Wang, T. Jiang, "Complexity of Multiple Sequence Alignment", Journal of computer biology. Journal Computational Biol. 2001;8(6): 615-23., pp. 337–348, 1994, doi: 10.1089/106652701753307511

[15] Menglin Cui, Rubin Bai, Zheng Lu, Xiang Li, Uwe Aickelin and Peiming Ge. "Regular Expression Based Medical Text Classification Using Constructive Heuristic Approach", IEEE Access Volume 7, pp. 147892-147903, 2019, doi:10.1109/access.2019.2946622

[16] Chaofan Tu, Ruibin Bai, Zheng Lu, Uwe Aickelin, Peiming Ge and Jianshuang Zhao. "Learning Regular Expressions for Interpretable Medical Text Classification", IEEE Access. pp. 1-4, 2019, doi:10.1109/cec48606.2020.9185650

Professor & Principal of SSBT's College of Engineering & Technology, Jalgaon. He is recognized PhD Guide in Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon. Professional Member in ACM, Life member of ISTE and CSI. His research interests are Wireless Sensor Networks and Security, Machine Learning, Blockchain and Natural Language Processing. He has published 30 research papers in reputed peer reviewed journals in addition to 12 papers in International Conferences to his credit. He is Senior Member in IEEE,

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 1376 - 1385

Mr. Dinesh Dagadu Puri, completed B.E from Walchand Engineering College, Sangli. and M.Tech. from DBATU, Lonere in Computer Science & Engineering. He is working as Assistant Professor in SSBT's College of Engineering and Technology since 2012. He is pursuing PhD in Computer Science & Engineering from Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon. His areas of interest are Machine Learning, Data Analytics and Natural Language Processing.

Mr. Akash Dnyandeo Waghmare, completed B.E. and M.E. in Computer Science & Engineering. He is Working as Assistant Professor in Department of Computer Engineering, SSBT's College of Engineering and Technology since 2013. He is pursuing his PhD in Computer Science & Engineering in Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon. He is a Life member of ISTE. His areas of interest are Machine Learning, Sentiment Analysis, Data Analytics and Blockchain.