EFFECTIVE DATA ALIGNMENT AND KEY EXTRACTION TECHNIQUES FOR GENERATION OF REGULAR EXPRESSIONS USING MACHINE LEARNING TECHNIQUES

[1]DINESH D. PURI, [2]Dr. G. K. PATNAIK

[1]Ph.D. Research Scholar, Department of Computer Engineering,
SSBT's College of Engineering & Technology Jalgaon.MH, India.

[2] Professor, Department of Computer Engineering,
SSBT's College of Engineering & Technology Jalgaon.MH, India.

Email: [1]ddpuri@gmail.com, [2]patnaik.girish@gmail.com

ABSTRACT

Machine learning supervised classification plays a significant role in large text classification. Health care data contribute to generating privacy and security for a couple of years. Such electric records might take extensive data from storage devices, so it needs to optimize with some processing techniques. For generation of regular expression, data should be in structured form. The conversion of unstructured data in to structured form is done through modified data alignment and key extraction techniques. The Smith-Waterman method compares two sequences to find comparable review statements. Smith-Waterman method employs an evolutionary algorithm to find improved local alignments of pairs. It can identify the best local alignment using the supplied score system. The NLP process has been used to generate the keys from such large text. In this paper, we proposed a sequence alignment generation using Smith-Waterman (SW) algorithm and key extraction from generated sequence using Natural Language Processing (NLP) technique has used for effective regular expression building. The filtration techniques have been used to eliminate redundant features, and the Machine Learning (ML) algorithm has been used as post-processing for classification. The generated regular expression by the SW algorithm gives better classification accuracy using NLP and Machine leavening. Generated regular expressions are filtered using 100% precision as threshold. We applied various Machine learning algorithms out of which SVM gives highest accuracy.

Keywords: *Sequence Generation, Sentence Alignment, Key Extraction, Regular Expression, Classification, Smith Waterman Algorithm, NLP*

1. INTRODUCTION

The sequence alignment is the process to generate possible sequences using pair wise sentence embedding methods such as Smith Waterman algorithms. In that process each symbol of one string is aligned with (i.e. displays on the same column as) a symbol of another string, or the blank symbol '-'. An alignment matrix is obtained in this manner. In an alignment matrix, a column with two identical symbols signifies a match, whereas a column with two different (non-blank) symbols represents a mismatch, or an index operation. No two blank symbols may be aligned. Each operation has a weight. An alignment's score is the sum of the scores of the columns in the alignment matrix. It computes the greatest alignment score for a pair of sequences. The issue of multiple sequence alignment may be defined as a pair of sequence alignment. There are many scoring algorithms (e.g. sum of pair wise distances) allocating a weight to each column in multiple sequence alignment of k sequences. There are several methods for the NP-hard issue of multiple sequence alignment.

The sequenced alignment challenge is one of the most pressing concerns for academics in terms of developing an optimal system model that may aid in optimum processor and efficiency while introducing overhead expenses in terms of memory and time. The following sections explain several sequence generating approaches that are often used to generate sequences from genomic and health care data.

Sequence comparison

The sequence database searching from [3] is one of the most difficult and significant problems in bioinformatics, where the rapidly increasing volumes of genetic data accessible constitutes a regular challenge to manufacturers of software and hardware record searching and administration. Every 15 months, the size of the nucleotide database doubles. Despite the fact that computing resources have been rising exponentially for many years, the fast expansion of genetic sequence information may be outpacing the rise in computer power. When searching for sequences that are comparable to a given analysis sequence in a database, the search algorithm calculates an alignment score for each database. Using the best algorithm for selected databases is time-consuming.

Sequence alignment generation method

The fundamental and most essential effort in the subject of bioinformatics is examining biological sequence databases and determining similarities between protein and DNA sequences. The Needleman-Wunsch (NW) algorithm was used to tackle this problem. Because the Needleman-Wunsch technique compares two complete sequences, the processing time becomes unsustainable owing to the exponential growth rate and vast volume of biological genomic sequence.

Multiple Sequence Alignment

Simultaneous sequence alignment is a nonlinear control approach to replicate copies of the automata from [10], in which a balanced finite automaton is constructed from a defined regular expression in order to find data sequencing with the highest possible score. This method has been simplified by the researchers: 1) Define the issue as inhibited pairwise comparison for multiple sequences, as indicated by the standard statement. 2) Developed a reliable solution for when sequential agreements are required to include a certain known sequence of standard statements.

Multiple Sequence Alignment using Regular Expression Constrain

The Smith-Waterman method is a well-known multiple sequence alignment procedure. Introduce a divergence of the issue for furthermore this, especially the pattern matching limit multiple sequence alignment, in this article [11], and give a method for it. This method was suggested to solve the issue of requiring alignments to include a certain sequence of regular expressions.

Performance Improvement of The Smith-Waterman Algorithm

This research [15] intends to provide a novel technique to improving Smith-Waterman performance utilizing partly modified hardware. The numerically difficult element of the method is accelerated using bespoke hardware in this manner. Rather than having the full algorithm implemented in hardware. When a tiny section of the technique is implemented in hardware, it speeds up 35.82 times faster than the software version.

Aligning generation for two sequences

With O(NW) computing time and O(N) gap [2], two sequences may be aligned inside a defined diagonal band. Scores may be calculated using local and global methods. After determining the optimum local alignment in the band, a global alignment method is used to connect the two spots. To generate optimum scores for each sequence in a protein reference sequence, this approach reduces the time needed by 40%.

Global Alignment

The Needleman Wunsch method is the most used global alignment algorithm. This method determines the best part of the permanent between two Nucleotide sequence (database and query) throughout their whole length. Because these two sequences are linked together, matches and incompatibilities are simple to see. The minimum match is a number determined by the sequences' similarity. Comparisons are done using the lowest unit of significance, a pair of organic molecules from each protein. The global technique found to be inadequate for the new DNA analysis technologies, which compared shorter sequences to a single big sequence. For this reason, global harmonization is seldom employed anymore.

Parallel Implementation of SW algorithm

The Smith Waterman algorithm is parallelized in this publication [1]. This approach is much faster than sequential implementation, while maintaining the same degree of sensitivity. A horizontal rectangular comparison becomes O(mn) while a multiple sequence pair comparison becomes O(mnk). Thus, a cost-efficient comparison mechanism is urgently required! Parallel implementation reduces time complexity while keeping the same degree of sensitivity for this crucial activity.

The organization of this paper as section II describes a review of literature that contains various sequence generation and key extraction methods for generating regular expression by existing researchers. Section III describes multiple methods for sequence alignment generation, while section IV describes the implementation of the proposed model with system architecture. Section V demonstrates results and discussion with an experimental set of system, and the final section VI focus on the conclusion and future work of this research.

2 LITERATTURE SURVEY

We cover relevant work on clinical named entity identification, as well as different sentence alignment and key extraction approaches employed by various studies, in this part. The majority of writers employed NLP to handle data, whereas other mining approaches were used to extract features.

A naive RNN model can theoretically handle the preceding information for each phase. However, the vanishing gradient and expanding gradient difficulties in back propagation through time (BPTT) cause it to fail to acquire enough information from prior stages and manage long-term dependencies in practise [1]. In order to solve this problem, the LSTM model was built.

Since the late 1970s, Clinical Patient Guides have served as a formal resource for identifying and adapting best practises in the medical arena [2]. The transfer of theory into ordinary clinical practise, in the form of textual CPGs, is only achievable when the theory is turned into a machine-interpretable model. Many techniques to achieving this transition have been presented, with a particular emphasis on expressing and implementing knowledge on large health care data. Document generation model, supervised model, decision Tree (DT) based models are some of the ways that may be utilised to describe recommendations in an acceptable fashion [3]. Despite this, the present technologies for automatically converting CPGs into computer-readable format have a number of drawbacks. Because most recommendation phrases are not expressed in the IF condition > THEN action > structure, the main challenge here is effectively identifying and extracting recommendation sentences from textual material. As a result, a method for extracting recommendation phrases from CPG text is needed. The following are some of the methodologies that are pertinent to the suggested strategy.

Using language patterns and specified templates, R. Servan et al. [4] created an approach for formalising CPGs. The domain ontology was used to build the language templates. Pattern extraction, selection of key patterns from extracted patterns for the construction of an executable model, and model assessment were all part of their suggested technique. This strategy resulted in the creation of reusable guidance blocks/templates for the creation and formalisation of CPGs. However, in order to map the ideas and generate the template, this method requires a specific domain ontology. This rule-based technique employs a mix of UMLS language and semantic data. Each CPG statement, the authors believed, has a domain-dependent linguistic and semantic structure. By finding the condition-action pair of phrases, they developed a weighting coefficient (relevance rate) to extract the relevant assertions. Their suggested approach can identify the relevance of each statement to the therapeutic process in this way. The authors utilised a single guideline to find and extract 12 "if" statements and four "should" statements. They discovered that rules of type "if" are more likely to be detected than rules of type "should."

For the categorization of CPG statements, H. Hematialam et al. [5] used supervised learning models based on ZeroR, Naive Bayes, J48, and Random Forest. Additionally, domain dependence limitations were removed using Part-of-Speech (POS) tagging, and recommendation statements were detected using modifiers and regular expressions. "If," "in," "to," "for," "when," and "which" were the most frequently used modifiers. Later on, the detected suggestion statements were converted into a "if condition then consequences" structure for rule creation. The models utilised by the authors were one-shot models, which need retraining every time the training dataset changes.

W. Gad El-Rab et al. [6] developed a framework for the active distribution of CPGs as well as the automated extraction of information from them. Some of the tasks in their suggested framework are automated to decrease manual labour. The system employs an form less material administration architecture (UIMA) to identify medical ideas and takes a multi-step approach. The authors did this by conducting XML parsing, text cleaning, medical concept tagging, medical tag disambiguation, clinical context pattern identification, clinical context filtering, and clinical context mapping, among other information processing tasks.

S. Priyanta et al. [7] used rule-based and machine-learning methods to compare the categorization of sentence subjects. For rule creation, the authors utilised opinion patterns. They evaluated sentence subjectivity in Indonesian news to determine if a sentence was a subject or an objective. Two machine-learning models, a Naive-based classifier (NBC) and a multinomial Support Vector Machine, were used to obtain this classification (SVM). The results of the assessment and analysis revealed that the rule-based classifier performed better, with an accuracy of 80.36 percent, compared to 74.0 percent for SVM and 71 percent for NBC. The disparity in accuracy between rule-based and machine-learning algorithms has precluded the latter from being used in real-world applications.

Other fields, including the clinical domain, rely heavily on pattern-based techniques for NLP tasks such as opinion mining in languages like Persian [8] and Chinese [9]. For Persian opinion mining, [8] presented a hybrid framework of dependency language rules and deep neural networks. The framework uses linguistic principles to determine the text's polarity. If no rule is present to trigger for an unknown occurrence, the framework uses a neural network model to classify the data. Similarly, the [9] employs Chinese language grammatical rules as constraints for the Bi-LSTM model, which outperforms other deep-learning models like RNN and LSTM in Chinese sentiment categorization.

Pre-processing is a crucial phase in the information processing process, since it transforms raw input data into a cleaner version. This transformation has a significant impact on the data-driven decision modelling pipeline, and it consumes 50 to 80 percent of the computing time [10]. Pre processing ultimate goal is to convert incoming data into a format that can be compiled using automated knowledge mining methods. The purpose of Document Pre processing in this research is to break CPG materials into sentences. Three sub-steps are required to reach this aim. To begin processing, the Document Reader loads the textual Clinical Patient Guide (CPG) document into computer memory. Second, all empty lines are removed, and multiple spaces are replaced with a single space to achieve format alignment. Finally, the Sentence Extractor splits the text into sentences using the Natural Language Toolkit (NLTK) sentence tokenizer. Pattern Extraction Process and Sentence Classification components use the retrieved sentences to extract patterns and find recommended phrases.

3 RESEARCH METHODOLOGY

For text summarization applications, semantic segmentation has critical tasks. We describe techniques for cleaning text statistical models in this section, which removes implicit noise and allows for informative series of information. We also go over two popular strategies for extracting text features: Function and reinforced word embedding approaches.

3.1 Text Cleaning and Pre-processing

In this process stop words, misstatements, slang, and other needless words abound in most text and document data sets. Noise and superfluous features may degrade the performance of many algorithms, particularly based on probabilistic learning algorithms. We'll go over several strategies and methods for text scrubbing and which was before text data sets in this part.

Tokenization

Tokenization is a pre-processing technique that divides a stream of text into tokens, which may be words, phrases, characters, or other significant items [11]. The analysis of the syllables [10] is the primary purpose of this stage. Text categorization and text mining both need the use of a parser to handle the smart contracts of the texts.

Stop Words

Many terms in text and information retrieval are not significant enough to be employed in classification techniques, such as "a", "about", "above", "across", "after", "afterwards", "again", and so on. To deal with some of these terms, the most typical method is to eliminate them from manuscripts. [12].

Capitalization

To compose a sentence, text and documentation data points have a variety of caps. Differential capitalization may be a big challenge when identifying lengthy papers since they include numerous phrases. When dealing with irregular capitalization, the most typical solution is to convert all letters to lower case. This approach merges all words in a text or document into a single feature space, although it complicates the translation of certain terms (for example, "US" (United States of America) to "us" (pronoun)) [12]. These deviations may be accommodated using slang and abbreviation converters. [14].

Slang and Abbreviation

Other types of text abnormalities that are addressed in the pre-processing stage include slang and abbreviation. SVM stands for Support Vector Machine, and an abbreviation [15] is a condensed version of a word or phrase that contains the initial letters from the syllables. Slang is a subset of spoken or written language that has several meanings, such as "lost the plot," which effectively means "gone insane" [16]. Converting these terms into formal language is a frequent approach of dealing with them [17].

Noise Removal

Many unneeded characters, such as capitalization and special characters, may be found in most text and documentation data sets. Although critical capitalization and special characters are necessary for human comprehension of papers, they may harm categorization systems. [18].

Spelling Correction

Spelling correcting is a pre-processing step that may be skipped. Typos (short for typographical mistakes) are widespread in texts and documents, particularly in text data sets from social media (e.g., Twitter). This challenge has been addressed by several algorithms, strategies, and methodologies in NLP [19]. Researchers may use a variety of strategies and procedures, such as hashing-based and context-sensitive spelling variety of experimental [20], as well as misspelling correction utilizing the Trie and Damerau–Levenshtein proximity bigram [21].

Stemming

In NLP, a single word might exist in several forms (for example, singular and plural noun forms), all of which have the same semantic meaning. Stemmer is one way for combining various variants of a word into the very same feature space. Using linguistic processes like affixation (the insertion of affixes), text stemming alters words to produce alternative word form. The stem of something like the word "planning to study," for illustration, is "plan study”.

Lemmatization

Lemmatization is a natural language processing technique that substitutes a meaning of a word suffix with an alternative one or eliminates the suffix entirely to reveal the fundamental word form lemmas features.

3.2. Syntactic Word Representation

Many researchers have attempted to solve this issue using unique ways, yet many of these strategies have drawbacks. Technical genomic texts were used to develop a model in which the utility of adding syntactic and semantic information in the text representation for sentence selection was shown. The n-gram approach for feature extraction is another solution for the syntactic issue.

N-Gram

The n-gram method is a collection of n-words that appear in a text set "in that sequence." Although this is not a text representation, it might be used as a feature to represent one.

Syntactic N-Gram

This discusses syntactic n-grams, which are characterized by pathways in syntactic dependency or component trees rather than the document's linear arrangement.

3.3. Weighted Words

The most basic type of weighting word extraction of features is TF, in which each word is transferred to a number that represents the number of times it appears in the corpus. Word frequency is often used as a Boolean or logarithmically scaled weighting in methods that extend the findings of TF. Each content is converted into a vector (with the same length as the content) holding the prevalence of the words in that content in all weight words techniques. Although obvious, this method is constrained by the fact that some terms regularly used in the vernacular may predominate in such representations.

Bag of Words (BoW)

BoW for short, is a method of extracting text attributes for use in modeling, such as data mining algorithms. The method is straightforward and adaptable, and it may be used to extract information from texts in a variety of ways.

Term Frequency

The occurrence in document d reflects the amount of times a given word t appears. As a result, we can observe that when a term occurs in the text, it gets more significant, which is reasonable. We can use a column to represent the text in the bag of word models since the sequencing of terms isn't important. There is an entry for each individual phrase in the document, with the number being the term frequency.

Inverse Document Frequency

It mostly assesses the word's relevance. The main goal of the search is to find relevant records that match the requirement. Because tf considers all words to be equally relevant, the term frequency may be used to determine the weight of a term in a document.

4. PROPOSED SYSTEM DESIGN

The below Figure 1 describes an regular expression generation from health care dataset. The Drug Review dataset has collected from Kaggle.com that contains numerous attributes. The dataset contents some noise or null attributes using preprocessing we eliminate such values and normalized such instances or attributes. The sentence or sequence alignment has done using Smith waterman algorithm and Stanford NLP parser has used for extraction of keys. The keys should be unique or non-redundant. In below section we describe each phase in detail;

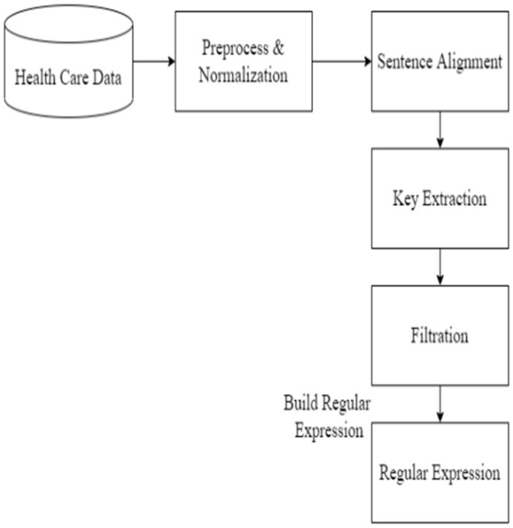

Figure 1 : Proposed system architecture

Data Collection: This module collects data from a variety of sources, including simulated and real-time situations in the health-care area. Some textual information and an illness description are included in the data. We gathered five distinct health-care datasets to employ in the proposed study.

Pre-processing and normalization: Misclassified instances were removed using a systematic sampling strategy throughout this step of token production. NLP methods were used to generate unique tokens and compute positive and negative phrases.

Regular Expression Generation: Feature selection approach was used to choose both positive and negative terms, and SW techniques were used to construct Regex code.

Regular expression generation utilizes the alignment results keys. The sequences of consecutive and aligned tokens are considered as a phrase, and an ordered list of aligned phrases as a key. In the text snippet “He has backpain from five years” and “he has sufferd from years”the aligned phrases are ‘he has’, ‘for’, and ‘years’. One of the keys is [‘he has’ ‘for’ ‘years’]. Given a set of aligned phrases, a number of keys may be generated, for example, [‘he has’], [‘for’ ‘years’], [‘he has’ ‘years’].

The keys generated using aligned phrases are passed on to the next steps to create regular expressions. Some of those regular expressions are then filtered out depending on their matching performance on the training data. If none of the expressions generated using the aligned phrases survived filtering, key extraction will iteratively add unaligned tokens. For example, assuming keys generated using ‘he has’, ‘for’, and ‘years’ did not produce any successful expressions, we will iteratively add by unaligned words in order to create new keys.

Consider the following two text snippets

S1: He suffered from two years by back pain

S2: He suffered by back pain

Alignment of these two text snippets will be as the following

He suffered for 2 years by back pain

He suffered by muscle pain

Aligned pharaes for S1: “He suffered”, “years”, “pain”

S2: “He suffered”, “pain”.

Keys can be formed from phrases of S1 and S2 are [He suffered], [years], [pain] and [He suffered], [for years], [pain]

These keys generated are passed to generate the regular expression.

For each key in two way regular expression can be generated. First method is with non-distance control and second method is with distance control.

Non Distance Control uses the phrases from key and uses (s/+/S)* pattern to permit any number of tokens in between phases.in the original alignment. Here s/ is white space character where /S is group of non-whitespace charater that is word. * represents zero or more than zero occurrence of words.

In Distance Control there is control on tokens to be inserted between aligned phrases.it uses the pattern (s/+/S)*{a,b} where a is minimum limit for token length and b is maximum limit for token length.

Regular Expression from Keys with non distance control we be as follows

He suffered (s/+/S)* years (s/+/S)* pain

He suffered (s/+/S)* pain

If distance control pattern is used then following will be regular expression

He suffered (s/+/S){1-3} years (s/+/S){2-4} pain

He suffered (s/+/S){1,3} pain.

Classification: Various supervised classifiers have been utilised for module training and certification, and they give full solutions similar to current heuristic approaches.

The classification algorithm is used for proposed system is given below

```
Input : Train data[], Test data[], weighted threshold, distance function DF[]
Output : Generated class label for test instance
1.  Read Test data from test matrix
        $Test\_Instance[] = \sum_{n=1}^{Testdata.len} (Att\ [n] \dots\dots Att[n])$
2.  Read train data from train matrix
        $Train\_Instance[] = \sum_{m=1}^{traindata.len} (Att\ [m] \dots\dots Att[m])$
3.  Calculate distance from both instance vector
    $Dist_{Corr(n,m)} = \sum_{n=1}^{Test\_Instance.len} (Att\ n) \sum_{m=1}^{Train\_Instance.len} (Att\ [m])$
4.  If$(Dist_{Corr(n,m)} = Null)$
5.  Failed classification
6.  Return $Dist_{Corr(n,m)}$
```

this is final class with distance weight with training instance

5. RESULTS AND DISCUSSIONS

After successfully implementation of proposed system evaluate the result with various experiment analyses with specific operating environment. Windows environment has used with, i7 processor, RAM is 12 GB and Graphics card with hard disk of 1TB. For the execution of the program, IDE (Netbeans 8.0) is used and it is implemented in Java (JDK 1.8). The below Table 1 describes an dataset description with training and testing instances.

Table 1 : Dataset Description

Dataset	Attributes	No. of instances
Training data	Depression	600
	Birth Control	550
	Pain	940
	Bipolar Disorder	1150
	Weight Loss	890
Testing dataset	Depression	240
	Birth Control	205
	Pain	380
	Bipolar Disorder	450
	Weight Loss	385

Here below, Table 2 describes dataset information with the number of attributes used in training as well as testing the dataset. In the attributed 2 “comment,” we proceed for sequence alignment as well as key extraction and finally, regular expression generation.

Table 2 : Attribute Description Of Dataset

Id	Attribute name	Attribute type
1	Comment _id	Numeric
2	Comment	String
3	Class	string

Figure 2 demonstrates the no. of regular expression generated using the proposed SW algorithm and NLP techniques. After applying the filtration algorithm, we evaluate the count of induced regular expression, which describes below Figure 2.

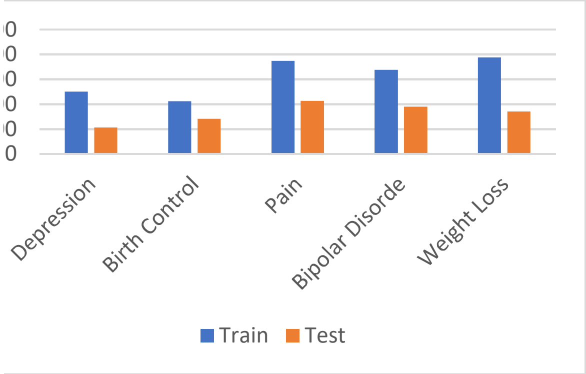

Figure 2 : No. Of Regular Expression Generated By Training And Testing Dataset

After successfully generating regular expressions, we use a supervised machine learning classification algorithm to classify those sets. The Weka 3.7 API was used to classify regular expressions. We use three machine learning algorithms to build diverse data chunks for system categorization. Three cross-validation processes were utilised for data splitting: 5-fold, 10-fold, and 15-fold. Lessons learned from supervised learning approaches are shown in Figure 3.

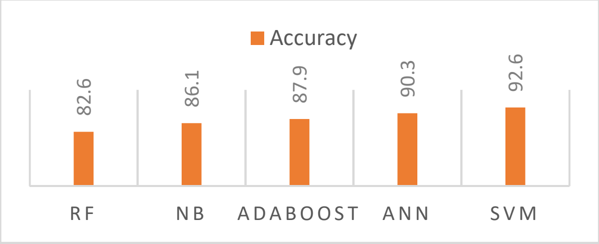

Figure 3 : No. Of Regular Expression Generated By Training And Testing Dataset

As a result of the research, many tests were carried out, each with a different algorithm, a single technique, and a combination of different approaches parameter updated configuration. The combination of SVM algorithms outperforms a single algorithmic solution. Existing system with the traditional approach gives 87.5% accuracy [31]. On large datasets, the prediction model computed and provided accuracy graphs that yielded almost 92.6% accuracy.

6. CONCLUSIONS

We offer a novel dynamic programming solution for pair wise comparison and key generation techniques that are bound by regular expressions.

The alignment of key sequences with the requirement that optimum alignment incorporates a specific sequence of motifs is a common application of this issue. We suggest changing the terms such that the section of the alignment that meets the restriction does not count toward the alignment's overall score. The calculations in our approach are split into two levels, each of which corresponds to a section of the nonlinear control matrix for conventional multiple sequence alignment. The overall size (volume) of these layers affects the performance of our method. For the regular expression limited multiple sequence alignment issue, our technique is more efficient than earlier solutions. In comparison to the standard (unconstrained) multiple sequence alignment, our technique not only takes no more time and space, but also outperforms it. To perform optimum solution for sequence alignment and key extraction method generate effective expression using deep learning technique.

7. FUTURE WORK

The optimization of regular expression can be done by applying various filtering techniques and same regular expressions can be used for classification. We can use deep learning to enhance the results.

REFERENCES:

[1] Pascanu, R, Mikolov, T, Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning (ICML 2013), Atlanta, GA, USA, 16–21 June 2013, pp. 1310–1318.

[2] Jacobsen, P.B. Clinical practice guidelines for the psychosocial care of cancer survivors: Current status and future prospects. Cancer 2009, 115, 4419–4429.

[3] Peleg, M. Computer-interpretable clinical guidelines: A methodological review. J. Biomed. Inform. 2013, 46, 744–763.

[4] Serban, R, ten Teije, A., van Harmelen, F., Marcos, M., Polo-Conde, C. Extraction and use of linguistic patterns for modelling medical guidelines. Artificial Intelligence Med. 2007, 39, 137–149.

[5] Hematialam, H., Zadrozny, W. Identifying condition-action statements in medical guidelines using domain-independent features. arXiv 2017, arXiv:1706.04206.

[6] Gad El-Rab, W., Zaïane, O.R., El-Hajj, M. Formalizing clinical practice guideline for clinical decision support systems. Health Inform. J. 2017, 23, 146–156. [PubMed]

[7] Priyanta, S., Hartati, S., Harjoko, A., Wardoyo, R. Comparison of sentence subjectivity classification methods in Indones an News. International Journal of Computer Science Information Security 2016, 14, 407.

[8] Dashtipour, K., Gogate, M., Li, J., Jiang, F., Kong, B., Hussain, A. A hybrid Persian sentiment analysis framework: Integrating dependency grammar based rules and deep neural networks. Neuro computing 2020, 380, 1–10.

[9] Lu, Q., Zhu, Z., Xu, F., Guo, Q. Chinese Sentiment Classification Method with Bi-LSTM and Grammar Rules. Data Analytics Knowledge Discovery. 2019, 3, 99–107.

[10] HaCohen-Kerner, Y., Miller, D., Yigal, Y. The influence of preprocessing on text classification using a bag-of-words representation. PLoS ONE 2020, 15, e0232525. [PubMed]

[11] Verma, T., Renu, R., Gaur, D. Tokenization and filtering process in RapidMiner. International Journal of Applied Information System 2014, 7, 16–18.

[12] Aggarwal, C.C. Machine Learning for Text, Springer: Berlin/Heidelberg, Germany, 2018.

[13] Saif, H., Fernández, M., He, Y., Alani, H. On stopwords, filtering and data sparsity for sentiment analysis of twitter. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland, 26–31 May 2014.

[14] Gupta, V., Lehal, G.S. A survey of text mining techniques and applications. Journal of emerging technology. Web Intell. 2009, 1, 60–76.

[15] Dalal, M.K., Zaveri, M.A. Automatic text classification: A technical review. International Journal of Computer Application. 2011, 28, 37–40.

[16] Whitney, D.L., Evans, B.W. Abbreviations for names of rock-forming minerals. Am. Mineral. 2010, 95, 185–187.

[17] Helm, A. Recovery and reclamation: A pilgrimage in understanding who and what we are. In Psychiatric and Mental Health Nursing: The Craft of Caring, Routledge: London, UK, 2003, pp. 50–55.

[18] Dhuliawala, S., Kanojia, D., Bhattacharyya, P. SlangNet: A WordNet like resource for

English Slang. In Proceedings of the LREC, Portorož, Slovenia, 23–28 May 2016.

[19] Pahwa, B., Taruna, S., Kasliwal, N. Sentiment Analysis-Strategy for Text Pre-Processing. International Journal of Computer Application 2018, 180, 15–18.

[20] Mawardi, V.C., Susanto, N., Naga, D.S. Spelling Correction for Text Documents in Bahasa Indonesia Using Finite State Automata and Levinshtein Distance Method. EDP Sci. 2018, 164.

[21] Dziadek, J., Henriksson, A., Duneld, M. Improving Terminology Mapping in Clinical Text with Context-Sensitive Spelling Correction. In Informatics for Health: Connected Citizen-Led Wellness and Population Health, IOS Press: Amsterdam, The Netherlands, 2017, Volume 235, pp. 241–245.

[22] Mawardi, V.C., Rudy, R., Naga, D.S. Fast and Accurate Spelling Correction Using Trie and Bigram. TELKOMNIKA (Telecommunication Computation Electronic Control) 2018, 16, 827–833.

[23] Spirovski, K., Stevanoska, E., Kulakov, A., Popeska, Z., Velinov, G. Comparison of different model's performances in task of document classification. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, Novi Sad, Serbia, 25–27 June 2018, p. 10.

[24] Singh, J., Gupta, V. Text stemming: Approaches, applications, and challenges. ACM Computational Survey. (CSUR) 2016, 49, 45.

[25] Sampson, G. The'Language Instinct'Debate: Revised Edition, A&C Black: London, UK, 2005.

[26] Plisson, J., Lavrac, N., Mladeni´c, D. A rule based approach to word lemmatization. In Proceedings of the 7th International MultiConference Information Society IS 2004, Ljubljana, Slovenia, 13–14 October 2004.

[27] Korenius, T., Laurikkala, J., Järvelin, K., Juhola, M. Stemming and lemmatization in the clustering of furnish text documents. In Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, Washington, DC, USA, 8–13 November 2004, pp. 625–633.

[28] Caropreso, M.F., Matwin, S. Beyond the bag of words: A text representation for sentence selection. In Conference of the Canadian Society for Computational Studies of Intelligence, Springer: Berlin/Heidelberg, Germany, 2006, pp. 324–335.

[29] Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L. Syntactic dependency-based n-grams as classification features. In Mexican International Conference on Artificial Intelligence, Springer: Berlin/Heidelberg, Germany, 2012, pp. 1–11.

[30] Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. J. Doc. 1972, 28, 11–21.

[31] Duy Duc An Bui, Qing Zeng-Treitler Learning regular expressions for clinical text Classification, Bui DDA, et al. J Am Med Inform Assoc 2014;21:850–857. doi:10.1136/amiajnl-2013-002411