Decision Support Complaint Prioritization System using a Statistical Multi-Method Algorithmic approach

Research · April 2023

DOI: 10.13140/RG.2.2.19079.98724

Decision Support Complaint Prioritization System using a Statistical Multi-Method Algorithmic approach

Hardik Srivastava,[a)] Mayank Jha,[b)] and T. Karthick[c)]

Data Science and Business Systems, SRM Institute of Science and Technology

[a)]hs9644@srmist.edu.in
[b)]mj9904@srmist.edu.in
[c)]Corresponding author: karthict@srmist.edu.in

Abstract Different organizations offer a variety of online grievance systems for people to address their grievances. However, given the volume and variety of complaints, there is a need to streamline and standardize the entire process while redressing them appropriately. Although these methods may address issues for most people, they do not redress problems faced by children who are subject to abuse and exploitation. Currently, no system implementation enables people to report cases of child labor to the concerned authorities digitally. In the wake of this problem as an implication, we developed a prototype of a GPS-based AI-Powered Android Application which leverages our proposed algorithm and effectively solves this problem. In this paper, we present a Multi-Method algorithmic approach to prioritize complaints using techniques like Text Mining, Sentiment Analysis, and Sentence Polarity Classification based on their content by assigning a redressal or a prioritization value to the case using a Decision Support System. We propose using a weighted average of the grievance sentiment, similarity measure, and the grievance form to find the priority category, amongst Severe, High, Medium, or Low, that the complaint will fall into. This prevents serious complaints from going unheard or even minor complaints from overwhelming the system. Our algorithm uses Machine Learning to rank grievances based on their priority and criticality with an accuracy of 90% and an AUC-ROC score of 0.930. This helps the needy and serious victims to receive assistance and support more quickly and appropriately.

Introduction

A grievance is a formal complaint made by someone experiencing or has encountered a problem at work. Grievance systems assist people in submitting formal complaints about various subjects to the appropriate authorities to have their issues addressed. However, there is a need for an effective methodology to make the current grievance redressal systems for children more reliable and accurate to prevent them from child labor and exploitation and provide timely aid. The use of online complaint redressal mechanisms by several organizations has made filing complaints quite simple [1] [2]. However, there are several disadvantages to the digital grievance system. It became more difficult for the authorities, given the current ecosystem, to resolve the issues adequately and effectively on time as the number of instances grew. Critical complaints are also unheard of as a result of all these problems. To stay ready for complaint redress, this opens up the possibility of developing an ecosystem through improved data modeling, design, data access, and analysis.

In this research, we propose a novel framework for text mining that combines Knowledge Discovery from Databases (KDD) [3] with Information Extraction (IE). The proposed methodology employs a modeling approach where text mining and sentence sentiment analysis [4] techniques are applied to a custom-curated child labor complaint dataset consisting of 42650 rows of processed data. The description of the child's physical state being reported is stored as the complaint data, and the person reporting the case is assigned a unique complainant identifier. A severity score is determined after analyzing the polarity and sentiment of the case's description data to create an efficient case prioritization mechanism.

We have implemented the DistilBERT model [5] in TensorFlow after fine-tuning it on our custom complaint dataset for the sentiment analysis task. To further prioritize and address similar and redundant complaints, we propose using priority weights [6] which are determined based on the similarity measure of each case. Sentiment analysis is used to determine the potential severity of a given complaint. The algorithm detects and analyses the content that contains greater grief and suffering, and that demands prompt attention and action. In addition to the content, the form of child exploitation, like Slavery, Trafficking, Debt Bondage, Sexual Exploitation, etc., plays an essential role in determining the priority weight for a particular case. A combined representation of the above sub-modules in the form of a weighted average [6] helps to rank grievances based on criticality and prioritizes them by their severity level, i.e., Severe, High, Medium, or Low. In contrast to Medium and Low Priority complaints, this approach aids in the decision-making process for resolving Top Priority concerns within a specific time frame. The authorities can determine which reported case from a list of instances ought to be addressed first based on the severity redressal score. Using the procedure described above, we produce a sorted list of complaints according to the priority of their resolution. It will take less time to prioritize and take necessary actions using this technique, and several complaints may be investigated daily impacting multiple lives.

This software implementation functions as an active tool that integrates an instant follow-up mechanism to act on the information obtained through efficient monitoring rather than passively ingesting data for statistical reasons. Our technique can provide good accuracy levels with less manual interpretation and intervention because there is no manual opinion involved.

The structure of the paper is as follows. Following the introduction, we provide a detailed description of our proposed methodology, the datasets used in our entire system implementation, formulation of the problem, and the multi-method algorithm architecture. Later in the paper, we talk about the various experiments that were carried out including the fine-tuning tasks and the training specifications. In the end, we talk about our results and discussions. The potential research directions to advance the field are covered in the Conclusion.

Materials and Methods

The concepts that guided the development of the system implementation are discussed in this section. The Grievance Redressal Systems may leverage this model architecture to implement efficient case monitoring strategies and prompt follow-up processes. Classifying complaints and giving them a severity redressal value or a priority value will enable the decision-maker to decide which complaints should be addressed first using the priority score. For cases involving severe forms and risks of child labor, it becomes really important for the authorities to redress them at the earliest than the ones having low priority. Figure 1 illustrates a schematic representation of our overall system implementation. In the system initialization phase, the necessary packages are added, along with the import of the dataset. For every complaint, a sentence-by-sentence sentiment score is computed, and its polarity is examined using text mining techniques.

In the next phase, each complaint's status is established after analyzing the sentence polarity based on its sentiment score, which is determined using the DistilBERT Deep Learning model [5]. There are four main bucket types: 'Low Priority', 'Medium Priority', 'High Priority', and 'Severe'. The case is classified as "Low Priority" if the priority value is higher than 0. The case is classified as "Medium Priority" if the priority value is equal to 0. Cases are classified as "High Priority" if the priority value is less than 0. However, if the priority value is below a certain pre-determined threshold, here -2, the child labor case is classified as "Severe". The complainant determines the form of child labor while he/she is reporting the complaint. To manage similar complaints reported by the same or different user, the grievances are first filtered based on the most relevant keywords generated from the newly raised complaint, and these filtered grievances are then evaluated by a similarity function. Here, the raised complaint's case description is compared to the case descriptions of the filtered grievances, and if the similarity exceeds or equals a certain pre-determined threshold, the grievance is identified as similar and given a priority weight. A user's complaint is flagged as spam if they submit it more than three times. To determine whether a grievance is similar to one that already exists, the system utilizes the grievance similarity module. Using the available user interface, the representative may delete it from the spam. The complaints are given priority sequence numbers according to the weighted average of sentiment score, similarity value, and the form of child labor. The algorithm outputs the priority category that each complaint belongs to, based on these priority scores. The priority sequence number is also produced as an output, which aids in the speedy resolution of critical concerns by sorting the complaints in ascending order of their severity redressal value.

Dataset

We created our complaint dataset by scraping child labor FAQs from complaint websites and numerous news articles that mentioned reports of child exploitation and child abuse at workplaces because of the non-availability of severity redressal grievance data concerning complaints raised for child labor. Table 1 shows the count of the total number of examples employed for training, validation, and testing tasks. We prepared a Python script using the Scrapy [7] framework to automate the data collection process. After certain unsuccessful attempts and hyperparameter adjustments, we managed to scrape 42769 rows of unprocessed raw data. Then, after applying a variety of pre-processing techniques like removing null data, filtering out complaints in languages other than English, removing special characters and punctuation, deleting case descriptions that contained advertisements and redirect URLs, and applying lemmatization [8] to each data point, we obtained 42650 rows of processed data that were suitable for fine-tuning our Sentiment Analysis model. Table 2 represents an overall count of examples in our dataset categorized by their form. Table 3 shows certain examples of pre-processed case descriptions from our dataset that we scraped from online websites and news articles.

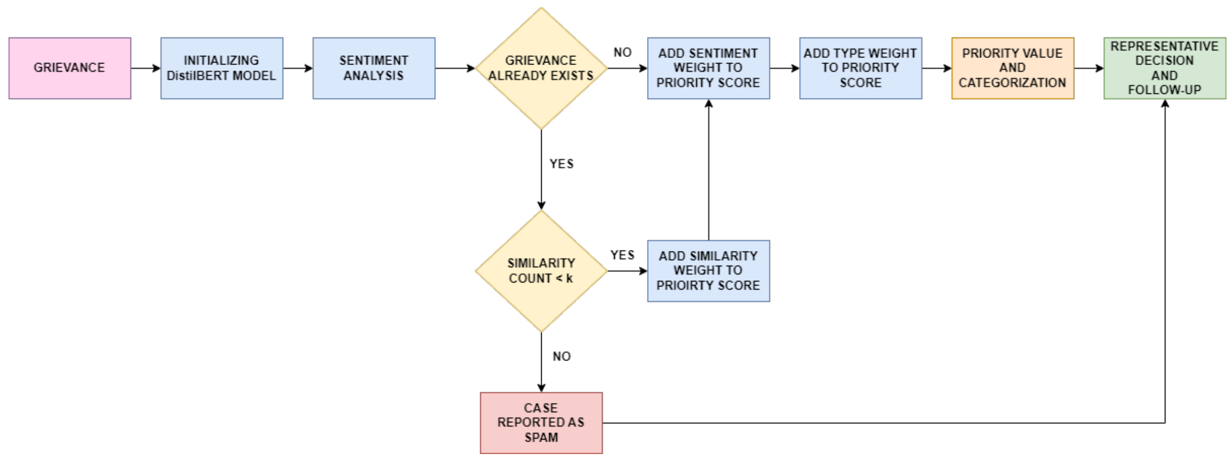

Figure 1: Representation of our overall System Implementation

Table 1: Representation of our dataset statistics

	#examples
Training set	29855
Validation set	8530
Testing set	4265

Problem Formulation

Now we formally propose our formulation of the problem. Our problem is split into two parts. The initial declaration involves defining our priority categories, namely Severe, Top, Medium, and Low. Assume the input complaint to be X. The first part of our problem deals with calculating a sentiment score using the DistilBERT Language model [5]. The model is fine-tuned for this task using our manually curated child labor grievance dataset, which contains rows of processed data on child grievances, particularly the physical state descriptions as described by the complainant. Upon each interaction with the language model, there is a numeric response between 0 and 1 as output. Let's consider this as X_S. Now for the second part, our priority algorithm first assigns a weight w_1 to X_S based on the weightage as discussed in the following sub-section *Algorithm*. Let's consider this as $X_1^{w_1}$. We then calculate the similarity value for X using BERT Sentence Tokenizer [9] [10] and assign a similarity weight w_2 to it. Let's consider this as $X_2^{w_2}$. Another type category weight w_3 is assigned to the final priority value based on the form of child labor. Let's consider this output as $X_3^{w_3}$. The final priority value Y is calculated by taking the weighted average of $X_1^{w_1}$, $X_2^{w_2}$, and $X_3^{w_3}$. Following is the equation (1) that describes the calculation of our priority value Y. Based on this value, a priority category is assigned to X. Our goal is to create a system implementation that automatically learns from these grievance data in a way that, given a test sequence $x \in X$ that hasn't yet been encountered, we can assign a priority value $\tilde{y}$, where $\tilde{y}$ determines which priority category the sequence x will fall into.

$$Y = \frac{\sum_{i=1}^{n} X_i^{w_i} w_i}{\sum_{i=1}^{n} w_i} \tag{1}$$

Table 2: Representation of the overall count of case descriptions in our dataset categorized by their form

Form	#examples
Debt Bondage	14417
Slavery	11141
Sexual Exploitation	9833
Trafficking	5069
Forced Labor	2190

Table 3: Examples from our dataset showing different types of pre-processed case descriptions of reported child labor categorized by their form as extracted from online complaint websites and news articles using web scrapping techniques.

<table>
<tr><th>S.No.</th><th>Form</th><th>Case Description</th></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Debt Bondage</td><td>The children were working in hazardous conditions as bonded labor in bakery units kharat machine units and auto center units of Alipur area of North Delhi district One child was rescued from a residential place where he was working as domestic help the DCPCR said in a statement</td></tr>
<tr><td>Since the last eight months twelve year-old Karan from Parbhani district in Marathawada region of Maharashtra wakes up at sunrise and for the next eight hours lugs bricks and cement to earn not more than three hundred per day</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Slavery</td><td>He had been made to work sixteen-seventeen hours a day with only a small lunch break The owner would supervise every phone call home Even a whiff of a complaint to the parents would yield beatings</td></tr>
<tr><td>Rajesh Sah of Kharauna village was lured around the time the pandemic began with five hundred rupees and the promise of three thousand rupees a sum he imagined to be considerable. Four other boys from the village were going with the man a known trafficker from a neighboring village Three months later he was not paid a penny</td></tr>
</table>

Algorithm

Our proposed Multi-Method algorithm takes into consideration multiple factors while calculating the severity redressal value (priority sequence number) and the priority category. The combined output of the multiple modules from our model architecture decides what value should be assigned to the raised complaint. *Algorithm 1* represents the algorithm for calculating the sentence polarity value using the DistilBERT model. *Algorithm 2* shows the algorithm for how the sentiment score that is calculated in *Algorithm 1*, the similarity measure of the raised complaint, and the form of child labor play a role in deciding the severity value for a case. Lastly, *Algorithm 3* represents the algorithm for finding out the category of the raised complaint based on the weighted average calculated in *Algorithm 2*. Following is a breakdown of the relative priority assigned to each component that affects the overall priority score:

- **Sentiment Score:** A grievance's level of negativity makes up 45% of its overall importance.
- **Similarity Measure:** A maximum of 35% of the similarity value affects the overall priority score.
- **Form:** Amounting to 20% of the overall priority are the priority weights assigned to the form of child labor.

Sentiment Analysis

The DistilBERT Transformer model is employed to produce the sentiment value of sentiment analysis. The customer complaint dataset we produced was tokenized using TensorFlow's BertTokenizer. The new complaint is fed into the model, which then generates a score between 0 and 1 using a Sigmoid Activation function [11]. This score represents how polarised the complaint is on a scale from negative (score of 0) to positive (score of 1). The final weight is decided by the algorithm and score.

Algorithm 1 Finding the Sentiment Score of the Grievance using DistilBERT model

```
procedure CALCSENTIMENT(grievanceData)
    Initialize Model and Load Weights
    Create Table ["ComplainantID", "Grievance", "SentimentScore"]
    for string(i) = 1 : len(grievanceData) do
        grievance ← grievanceData[i]["grievance"]
        sentimentScore ← sentiment(grievance, model)
    end for
    sentimentScoresData.csv ← Table["ComplainantID", "Grievance", "SentimentScore"]
    return scoresData
end procedure
```

Similarity Measure

We utilized TensorFlow's implementation of the BERT Sentence Tokenizer since it takes the text semantics into consideration while tokenizing them into vectors. The Cosine Similarity metric was used to calculate the similarity between the word embedding vectors from the tokenizer.

Type Categorization

There are various forms of child labor like Debt Bondage, Slavery, Sexual Exploitation, Domestic Labor, etc. The priority weights for each case vary depending upon the form reported by the complainant. For e.g., cases of Sexual Exploitation should be redressed at the earliest and should fall under the Severe or Top Priority category. Hence, the type amounts to a maximum of 20% of the overall priority.

Algorithm 2 Finding the Priority Value of the Grievance based on Sentiment, Similarity, and Form

```
procedure CALCPRIORITYVAL(sentimentScoresData)
    Create Table ["ComplainantID", "Grievance", "SentimentScore", "PriorityValue"]
    Initialize priority and priorityRange
    for string(i) = 1 : len(sentimentScoresData) do
        grievance ← sentimentScoresData[i]["Grievance"]
        sentimentScore ← sentimentScoresData[i]["SentimentScore"]
        sentimentPriority ← sentimentPriority + (1 − sentimentScore) ∗ sentimentWeight ∗ priorityRange
        grTypeScore ← getGrTypeScore(type)
        grTypePriority ← grTypePriority + (grTypeScore)/(totalTypes) ∗ typeWeight ∗ priorityRange
        kwsExtracted ← extractKwsFromGr(grievance)
        similarGrs ← getSimilarGrs(kwsExtracted)
        for string(complaint) = 1 : len(similarGrs) do
            if getSimilarCount(complaint) <= 5 then
                simPriority ← simPriority + similarityWeight ∗ priorityRange
            end if
        end for
    end for
    prioirty ← calcWghtAverage(sentimentPriority, grTypePriority, simPriority)
    pValueData.csv ← Table["ComplainantID", "Grievance", "SentimentScore", "PriorityValue"]
    return pValueData
end procedure
```

Severity Redressal Value

The final severity redressal value or the priority value is determined by taking a weighted average of the outputs from the modules explained in sub-sections *Sentiment Analysis*, *Similarity Measure* and *Type Categorization*. This value determines which category bucket out of the four the reported complaint will fall into.

Algorithm 3 Finding the Priority Category of the Grievance based on the Priority Value

procedure FINDCATEGORY($pValueData$)
  Create Table ["ComplainantID", "Grievance", "SentimentScore", "PriorityValue", PriorityStatus]
  $threshold \leftarrow -2$
  for $string(i) = 1 : len(pValueData)$ **do**
    $x \leftarrow pValueData[i, "PriorityValue"]$
    if $string(x) > 0$ **then**
      $"LowPriority" \leftarrow pValueData[i, "PriorityStatus"]$
    else if $string(x) < 0$ **then**
      $"HighPriority" \leftarrow pValueData[i, "PriorityStatus"]$
    else if $string(x) < threshold$ **then**
      $"Severe" \leftarrow pValueData[i, "PriorityStatus"]$
    else
      $"MediumPriority" \leftarrow pValueData[i, "PriorityStatus"]$
    end if
  end for
  $grPriorityData.csv \leftarrow Table["ComplainantID", "Grievance", "SentimentScore", "PriorityStatus"]$
  return $grPriorityData$
end procedure

Experiments

Pre-Processing

We utilized TensorFlow's implementation of BERTTokenizer [10] for tokenizing all complaint data in our dataset and WordNet [8] Lemmatizer from Scikit Learn to convert all tokens in a sentence to their root form. Lemmatization was preferred over Stemming as we wanted the root words to retain their context.

Fine-Tuning

We leveraged DistilBERT Language Model and tweaked it with additional training data to make it perform our Sentiment Analysis task. We express the input embeddings as the sum of the token embeddings, the segmentation embeddings, and the position embeddings in order to fine-tune the sentiment analysis model on our complaint dataset [12]. The pretraining head of the model and replaced with a classification head. When fine-tuning, only the classification layer weights $W \in R^{N \times H}$, where N represents the number of output labels, are added as additional parameters. We use a batch size of 16 and fine-tune for 100 epochs over the data for the sentiment analysis task. We choose the best fine-tuning learning rate of 2e-5 for each task on the validation set. With random restarts, various fine-tuning data shuffling and classifier layer initializations are carried out but the same pre-trained checkpoint is used each time for evaluation [13]. Our model achieved an accuracy of 90% and an AUC-ROC [14] score of 0.930 on the test set.

Results and Discussions

In this paper, we discussed applying an effective prioritization mechanism on grievances involving instances of child labor. Our proposed Multi-Method algorithm can be leveraged by any existing grievance redressal system to incorporate an immediate follow-up mechanism by assigning a priority status to complaints thereby sorting them in ascending order by their respective priority values. We also discussed several factors utilized by our algorithm that affect the calculation of the overall priority value.

We ran the algorithm on our entire dataset to check its efficacy. Table 4 summarizes our results and shows the priority categories with their count of grievances as outputted by our algorithm. Out of 42650 instances, the algorithm classified 6891 examples as "Severe", 11326 as "High Priority", 17475 as Medium Priority, and 6958 examples as Low Priority grievances. Using this technique, numerous child labor complaints can be solved in a timely manner by providing the respective authorities with an effective decision-making mechanism that lets them prioritize the Severe and Top Priority complaints first over the Low Priority ones in an orderly manner and allow them to initiate immediate follow-up mechanisms to solve the reported grievances.

Table 5 shows an overview of the various models and approaches that have been implemented for Grievance Redressal and Sentiment Analysis tasks.

Table 4: Representation of each Priority Category along with their respective count of grievances

Priority Category	#examples
Severe	6891
High Priority	11326
Medium Priority	17475
Low Priority	6958

Table 5: Comparative analysis between various models and approaches, that have been employed for Grievance Redressal and Sentiment Analysis tasks, and our proposed system implementation

Approach	Year	Algorithm	Task	Dataset
Indian Banks' Customer Complaints System [15]	2019	Random Forest, Naive Bayes, and TF-IDF	Sentiment Classification	Bank Customers Survey Data
Citizen Satisfaction and Prioritizing Their Needs [16]	2019	Clustering based on Frequency and Timeline	Sentiment Analysis	Citizens Complaint Data
Civil Complaints Management System [17]	2021	Machine Learning and TextBlob	Polarity Analysis	Consumer Complaint Dataset
AI-Driven Complaint Management System [18]	2021	Keyword Generation and Polarity Analysis using NLTK	Sentiment Analysis	Sentiment Treebank Dataset
Our Multi-Method Algorithm	2022	Weighted Average using Sentiment Analysis, Similarity Measure, and Type Categorization	Complaint Prioritization, and Polarity Analysis	Custom Complaint Dataset

Indian Banks' Customer Complaints System [15] talks about using Random Forest and Naive Bayes classifiers for the Sentiment Classification task. Similarly, Citizen Satisfaction and Prioritizing their Needs [16] proposes to use clustering techniques based on frequency and timeline using a Citizens Complaint dataset. Sayali Bhosale *et al.* [17] and Shreyas Shedge *et al.* [18] discusses calculating Sentiment scores using TextBlob and Keyword Generation techniques using NLTK. Some of the approaches that we drew comparisons with either introduced new approaches for existing Sentiment Analysis tasks or talked about using Sentiment Analysis in a way that can be used for Grievance Redressal. We instead propose an optimization in the existing methodology of redressing complaints for child labor by using a Multi-Method algorithm that acts as a prioritization mechanism for future complaints. It not only considers the sentence sentiment but also various other factors like the similarity measure and the form of grievance in a weighted fashion when calculating the priority.

Conclusion

Following the identification of possible improvements and the gaps in the current redressal systems, we have proposed a smart multi-method algorithm for grievance redressal systems to prioritize complaints and track and resolve them in an efficient and reliable manner. The system arranges the grievances in ascending order of their severity level or the cruciality of the case based on the priority sequence value. Prioritizing the complaints reduces the overall number of cases so that the authorities may concentrate more on the Severe and the Top Priority ones. This effective case prioritization mechanism is powered by a sophisticated algorithm that incorporates multiple methods and factors like sentence sentiment polarity, count of already reported grievances for measuring the similarity scores, and the grievance form-wise priority. In the end, the priority value is used to categorize the grievances into one of the four categories: Severe, High, Medium, or Low. We are certain that our proposed approach and the system implementation would serve as a form of prevention for exploited children and facilitate effective e-governance while offering people a positive dispute-resolution experience.

Acknowledgments

We wish to acknowledge the support of the School of Computing, SRM Institute of Science and Technology in offering suggestions and encouragement, and guiding us to carry out our research.

REFERENCES

1. Rana NP, Dwivedi YK, Williams MD et al (2015) Examining the success of the online public grievance redressal systems: an extension of the IS success model. Information Systems Management. 32(1): 39-59.
2. Gopakumar, "Transformative e-governance and innovation: challenges and impact of mass contact program of chief minister of Kerala 2013-15," 2016 IEEE International Symposium on Technology and Society (ISTAS), Kollam, India, 2016, pp. 1-4, doi: 10.1109/ISTAS.2016.7764277.
3. Matheus, Christopher & Chan, Philip & Piatetsky-Shapiro, Gregory. (1998). Knowledge Discovery in Databases. IEEE Transactions on Knowledge and Data Engineering.
4. Dang, N. C. (2020). Sentiment Analysis Based on Deep Learning: A Comparative Study. arXiv. https://doi.org/10.3390/electronics9030483
5. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv. https://doi.org/10.48550/arXiv.1910.01108
6. K. A. Djaballah, K. Boukhalfa and O. Boussaid, "Sentiment Analysis of Twitter Messages using Word2vec by Weighted Average," 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, 2019, pp. 223-228, doi: 10.1109/SNAMS.2019.8931827.
7. Asikri¹, M & Krit, S & Chaib, Hassan & Salah-ddine, Krit. (2020). Using Web Scraping In A Knowledge Environment To Build Ontologies Using Python And Scrapy. European Journal of Translational and Clinical Medicine. 7. 433-442.
8. Khyani, Divya & B S, Siddhartha. (2021). An Interpretation of Lemmatization and Stemming in Natural Language Processing. Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology. 22. 350-357.
9. Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv. https://doi.org/10.48550/arXiv.1810.04805
10. Mohebbi, H., Modarressi, A., & Pilehvar, M. T. (2021). Exploring the Role of BERT Token Representations to Explain Sentence Probing Results. arXiv. https://doi.org/10.48550/arXiv.2104.01477
11. Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation Functions: Comparison of trends in Practice and Research for Deep Learning. arXiv. https://doi.org/10.48550/arXiv.1811.03378
12. Z. A. Guven, "Comparison of BERT Models and Machine Learning Methods for Sentiment Analysis on Turkish Tweets," 2021 6th International Conference on Computer Science and Engineering (UBMK), Ankara, Turkey, 2021, pp. 98-101, doi: 10.1109/UBMK52708.2021.9559014.
13. X. Fan, X. Li, F. Du, X. Li and M. Wei, "Apply word vectors for sentiment analysis of APP reviews," 2016 3rd International Conference on Systems and Informatics (ICSAI), Shanghai, China, 2016, pp. 1062-1066, doi: 10.1109/ICSAI.2016.7811108.
14. Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, 30(7), 1145-1159. https://doi.org/10.1016/S0031-3203(96)00142-2
15. Krishna, Gutha & Ravi, Vadlamani & Reddy, Bheemidi & Zaheeruddin, Mohammad & Jaiswal, Harshal & Teja, P & Gavval, Rohit. (2019). Sentiment Classification of Indian Banks' Customer Complaints. 10.1109/TENCON.2019.8929703.
16. Ghodousi M, Alesheikh AA, Saeidian B, Pradhan B, Lee C-W. Evaluating Citizen Satisfaction and Prioritizing Their Needs Based on Citizens' Complaint Data. Sustainability. 2019; 11(17):4595. https://doi.org/10.3390/su11174595
17. Bhosale, Sayali & Patankar, Sonali & Kadam, Kshitija & Dhere, Rujuta & Desai, Prof. (2021). Survey on Civil Complaints Management System by Using Machine Learning Techniques. International Journal of Advanced Research in Science, Communication and Technology. 606-610. 10.48175/IJARSCT-1449.
18. Wadkar, Pratik & Raorane, Atreya & Bushra, Shaikh & Shedge, Shreyas. (2021). AI-Driven Complaint Management System. SSRN Electronic Journal. 10.2139/ssrn.3866078.