Graph based Keywords Extraction from Crime Data

Priyanka Das, Anuvab Munshi,
Annwesha Naha, Adrija Dey, Sriparno Chakraborty
Dept. of Computer Application and Science
Institute of Engineering & Management, Kolkata
University of Engineering and Management, Kolkata,
Email: priyankadas700@gmail.com, anuvabmunshi@gmail.com,
annweshanaha2005@gmail.com, adrijadeyxi@gmail.com, chakrabortysriparno@gmail.com

***Abstract*—A lot of data that could be gathered to help solve and avoid crimes is never done so. The best ways for citizens and law enforcement personnel to report crimes are inadequate. The current email and text-based solutions only partially help witnesses recollect their memories. The proposed study acquires crime-related keywords from intricate police and witness narrative descriptions by employing natural language processing techniques and a graph-based strategy. Police investigators can quickly understand criminal scenarios with the use of criminal information extraction, saving them from having to read the entire report. In order to accomplish this, the proposed work has initially assessed the importance score of the words present in the crime reports and then a graph partitioning technique has been employed. The graph partitioning algorithm is applied repetitively to group all terms into clusters, each of which symbolizes a distinct component of crime.**

***Index Terms*—Keywords extraction, Graphs, Crime Analysis, Permutation Importance, Clustering**

I. Introduction

In India, crime is perpetrated every three seconds on average. Most crimes are now solved after investigators interview witnesses and victims, analyse the narrative reports, and synthesise the data. When data is collected via questionnaires or structured interviews, it is common to get missing, no-response, or partial no-response results. Interviewers may forget to record data, seek to complete an interview fast, or record data erroneously. Furthermore, police detectives must go through a large number of narrative reports to find key clues. Text reports might also be automatically analysed to gather pertinent crime-related information such as car names, addresses, narcotics, and people's identities, which would increase efficiency. Information extraction can be used to obtain criminal information. Witnesses and victims can report directly by establishing a crime information extraction system. Automatically extracting information, merging information from crime narrative reports, and giving a meaningful summary for police investigators will allow them to rapidly understand crime situations without having to read the entire report. In general, obtaining these documents is challenging due to confidentiality and privacy concerns. A good and usable criminal information extraction system should be very accurate regardless of the kind or origin of the information. By analysing crime corpora from various data sources, a rich and extensive lexicon can be built. The performance is measured by comparing the extraction of information from police, witness, and victim narrative reports. The widespread phenomenon of excessive unstructured crime data becomes increasingly visible. It takes a long time for the law enforcement agencies to get the information they need. Assessing such a massive quantity of reports can be made easier if we have a specific type of words (Keywords) that give us with the document's primary features, hypothesis, motif, and so on. Keywords or keyphrases are extremely valuable for analysing vast amounts of textual content diligently and quickly. This extraction of keywords or keyphrases is effective for a variety of other applications. Keywords or keyphrases are representative terms in a document that provide high-level content specification. Relevant keywords can serve as a succinct summary of a document, allowing us to conveniently organise and locate documents pursuant to their content. For crime data, mining pertinent keywords from crime information and delivering a meaningful summary to police detectives will allow them to solve crimes more quickly without having to read the complete report. Manually assigning keywords is time-consuming and costly, while automatic assignment is more efficient. To extract keywords from reports, an automated technique is required.

The additional part of this study procedure is laid out as follows: Section II covers the most relevant works in the literature. Section III explains the paradigm for categorising food items as either nourishing or non-nutritional. Section IV summarises and discusses the findings. Section V brings the work to a conclusion.

II. Related Work

Text mining relies heavily on keyword extraction. Keyword extraction can be done using multiple ways, spanning supervised and unsupervised machine learning, statistically driven, and linguistic actions. Automatically extracting keywords from text as text tags improves recommendation and keyword-based information retrieval. One such example is shown in [1], that demonstrates how keywords are extracted from text using word frequency and association features. This research provides a novel way to extracting keywords from text that incorporates factors like word frequency and association. The experiment findings reveal that the precision rate, recall rate, and F-measure all outperform TextRank and TF-IDF. An article [2] presents a deep neural network model for keyword extraction. Two extensions to the traditional LSTM model have been

suggested. To begin, a target center-based LSTM model (TC-LSTM) has been proposed to better use both the historic and contextual information of a given target word. This model learns to encode the target word while taking into account its contextual information. Second, the self-attention mechanism has been added to the TC-LSTM model, allowing the model to focus on informative portions of the linked text. Furthermore, a two-stage training strategy has been devised that makes use of large-scale pseudo training data. The experimental results suggest that the two-stage training procedure is very important for boosting the model's effectiveness. The Graph of Words representation, the statistical properties and structural integrity of the word graph, the variety of two approaches, and the node order in the word graph were all studied in an article published in [3] as part of the Graphical Keywords Extraction Technique. In order to derive some conclusions, a thorough analysis of GKET is carried out across multiple industries to highlight these numerous qualities. These deductions are employed to propose possible research directions for inter-disciplinary network science and natural language processing studies. Additionally, the experimental results are assessed in order to analyze Word Co-occurrence Networks for fifteen different languages, compare and contrast the current GKET across twenty-one different datasets, and look at the structure of WCN for different genres. In this paper, certain strong correspondences in distinct disciplinary techniques are identified for different dimensions, namely, GoW representation: 'Line Graphs' and 'Bigram Words Graphs'; feature extraction and selection utilising eigenvalues: 'Random Walk' and 'Spectral Clustering'. The other issues related to keywords extraction are also discussed in [4]. Another research reported in [5] proposes an improved approach for extracting TextRank keywords. The algorithm calculates the importance of words using the TF-IDF algorithm and the average information entropy algorithm, and then computes the comprehensive weight of words in the text based on the calculation results. The comprehensive weight of words improves the initial weight of the TextRank algorithm node and the node probability transfer matrix, and the weights of all nodes are iteratively calculated until convergence is reached. The weights of the nodes are organised to obtain the weight data pertinent to the words, after which the top N words choose to serve as the keywords. In a study [6], an architecture GraphIE has been implemented that acts on a graph expressing a wide range of connections between textual elements. The approach uses graph convolutions to transport information between connected nodes, resulting in a more complete representation that can be used to improve word-level predictions. GraphIE regularly outperforms the state-of-the-art sequence tagging model by a significant margin in three different tasks: textual, social media, and visual information extraction. Semantic Clustering TextRank, a semantic clustering news keyword extraction technique established on TextRank, has been put forward in [7]. In the initial step, the word vectors generated by the Bidirectional Encoder Representation from Transformers (BERT) model are clustered using k-means to reflect semantic clustering. The clustering findings are used subsequently to generate a TextRank weight transfer probability matrix. Finally, word graphs are recurrently calculated and keywords extracted. Latent semantic analysis is also done to extract keywords from documents in [8]. A semantic relation is a model for demonstrating the semantic similarity of two or more terms, exhibiting the relationship between words in a sentence and with other sentences. This technique extracts a set number of key phrases from documents to identify the text's primary content. Keywords are also extracted from academic literature knowledge graph in [9]. The Conditional Random Fields (CRF) model is a cutting-edge sequence labelling method that makes greater use of document properties. At the same time, keyword extraction can be viewed as string labelling. The experimental work in [10] proposes and implements keyword extraction based on CRFs. As far as we know, there has been no previous research into employing a CRF model for keyword extraction. Experimental results reveal that the CRF model surpasses other forms of machine learning such as support vector machines, multiple linear regression models, and so on in fulfilling the task of keyword extraction.

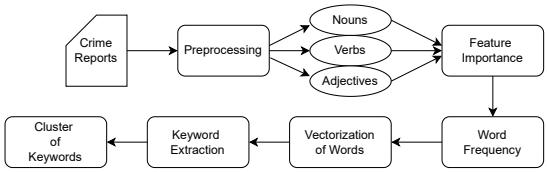

Fig. 1: Workflow diagram of the proposed work

III. Proposed Methodology

This section contains expert illustrations of the specified job. An approach for extracting keywords from crime data that is based on graph clustering is demonstrated in this study. Fig. 1 depicts the workflow diagram of the proposed approach. More information on the techniques is provided below:

A. Data Collection and Preprocessing

A handful of websites and a few Indian classified newspapers were used to extract data sets for the current inquiry. Information about crimes against women in Indian states and union territories can be found on these websites. Online versions of publications like "The Hindu", "The Times of India", and "The Indian Express" are also selected. Table I presents a broad summary of the data set.

TABLE I: Specifics of the experimental data

# Reports	# Sentences	# Tokens
5,929	46,805	523,320

As stop words could create noise while attempting to identify links between the words, they were eliminated from the data after it was collected during preprocessing. We have concentrated on extracting keywords from the data, therefore this is necessary. The sentences have then been tokenized into discrete parts, or distinct words, once stemming has been completed to yield only the root words. At last, we have completed the Parts of Speech (POS) Tagging and gathered

all nouns, verbs, and adjectives. In this case, we have taken into account verbs and adjectives in addition to nouns for the purpose of keyword extraction, on the grounds that these terms offer valuable context for the study of crime data. Table II provides a count of nouns, verbs and adjectives considered after the POS tagging. These are the features for the proposed work.

TABLE II: Count of nouns, verbs and adjectives

# Nouns	# Verbs	# Adjectives
267098	129754	77383

B. Calculation of Feature Importance

It is observed from the data given in Table II that the POS tagging has lead to generation of a large corpus of features and some are redundant among those words. Hence, a minimal subset of features are chosen by calculating their importance score. We incorporated the concept of permutation importance for determining the important features. We have employed the Random Forest Classifier that has been trained for this purpose, and we have recorded the accuracy score on the validation set. After that, we reshuffle one feature's values, make another prediction using the model, and determine the validation set's scores. The difference in the two scores is the feature importance. This technique has been repeated for all the nouns, verbs and adjectives. In this case, we make use of the permutation importance function that is included in the Scikit-learn package in Python. We shuffle each variable 20 times at random and determine the accuracy drop. This permutation importance has been done simultaneously for the nouns, verbs and adjectives. The details of the process is explained in Algorithm 1. After this importance score is obtained, a threshold of 0.5 has been set and the particular words having 0.5 or above the threshold has been considered for further processing. Table III provides the details after considering the threshold importance scores of the features.

C. Measuring Word Frequency

The term frequency (TF) measures the total number of times a word appears in a document and can be used to gauge a word's relevance overall. We have considered the word frequency because we agreed upon the fact that a word has a higher level of relevance to the topic of the document and a better chance of becoming a keyword if it appears more frequently in the document. In order to calculate word frequency, we took into consideration the total number of words in the document, or the number of times a word appears in the document, using the document length. The word frequency has been determined by the eq. (1).

$$wf(n) = \frac{count(n)}{doc\_len} \tag{1}$$

Here, $wf(n)$ be the frequency of word n, $count(n)$ gives the sum up of occurrences of word n in the document and doc_len is the length of the document.

Algorithm 1: Computation of feature (word) importance

Input: Dataset of crime reports R;
Output: The permutation importance score matrix S;
begin
  Design a matrix S for storing the permutation importance scores of the features or words present in data R based on the predictive model M;
  for *each feature (word)* $w_j \in R$ **do**
    for *each repetition* k *in* $1, 2, \cdots, K$ **do**
      Randomly shuffle column of j in data R to generate a new version of data $\tilde{R}_{k,j}$;
      Compute the score $S_{k,j}$ of features on model M on new data $\tilde{R}_{k,j}$;
      Compute the permutation importance for feature w_j as: $I_j = S - \frac{1}{K}\sum_{k=1}^{K} S_{k,j}$
    end
  end
  Return the matrix S with importance scores;
end

TABLE III: Count of nouns, verbs and adjectives after Permutation Importance

# Nouns	# Verbs	# Adjectives
160,258	64,877	48,691

D. Feature Representation by Glove Embeddings and Feature Similarity

Word vector representations can be obtained using the unsupervised learning method GloVe [11]. It starts by utilizing a provided corpus to generate a word-context co-occurrence matrix. Each entity in the matrix indicates the frequency with which two words appear together in a context window of a particular size. By reducing the dimensionality of the co-occurrence matrix, GloVe basically learns word vectors. Let $W = \{w_1, w_2, w_3, \cdots, w_n\}$ be the set of words chosen after the POS tagging. Then the similarity between two words w_i and w_j is measured using the Cosine similarity and Glove embedding, as depicted in eq. (2) where $\vec{w_i}$ defines word vector of w_i based on the Glove word embedding model and $||\vec{w_i}||$ defines the magnitude of vector $\vec{w_i}$.

$$Cosim(w_i, w_j) = \frac{\vec{w_i}, \vec{w_j}}{||\vec{w_i}||.||\vec{w_j}||} \tag{2}$$

We set the cosine similarity between terms w_i and w_j to [0, 1] as the normal.

E. Keyword Extraction Algorithm

This section covers the keyword extraction methodology in detail. The major goal of the proposed work is to identify keywords from crime data. The keywords are the most significant features here. The essential notations and preliminary steps utilized in our suggested graph based algorithm has been defined. We use the conventional notations from network theory to represent the set of vertices V and edges E,

respectively. Then, a complete weighted and undirected graph $G = (V, E, W)$ is constructed, where V the set of vertices is connected by set of edges E and W is the set of weights assigned to the edges. The sparse graph is represented as $G_s = (V, E_s, W_s)$. Here, the vertices are the words considered from Table III and the edges connecting the words have been assigned the weight which is reciprocal of the similarity score calculated using eq. (2). Therefore, higher the similarity among the words, lesser is the weight assigned to the edges. As we have a large number of features even after considering the permutation importance score, we have rendered the graph sparse based on a threshold associated with each vertex. For each node, $th(G)$ represents its average resemblance to other nodes in the graph, and the edge incident to this node is eliminated if its associated weight exceeds the average value. Here, the keyword extraction algorithm has been applied by combining the permutation importance score of the vertices or words present in the graph along with the frequency of that word measured using eq. (1) and Algorithm 1 respectively. The total weight considered for a vertex or a word is measured using eq. (3).

$$W(n) = wf(n) + I_j(n) \tag{3}$$

Next, for each vertex $v_i \in V$ in G_s, we have computed the total weight using eq. (3). The vertex having the highest weight has been extracted separately with all the edges incident on it as a partition G_1. Now, the vertices present in the partition G_1 has some weight. For the remaining graph $\tilde{G_s} = G_s - G - 1$, again we have followed the same process of selecting the vertex with highest weight. If there are numerous vertices with the same highest weight, we select the one with the highest degree. If numerous vertices have the same highest degree, then the tie is broken randomly [12]. Let the extracted partitions be g_h and g_i. The partition gets merged with the prior partition G_1 if and only if the resultant partition's total weight exceeds both g_i and g_h. Therefore, it may happen that there can be a single partition of the graph or two separate partitions. The partitions are basically the subgraphs containing the words which are considered as the keywords. The process will be repeated until and unless all the nodes and edges of the original complete weighted graph are exhausted. The subgraphs represent the cluster of the keywords. The complete process is described in Algorithm 2.

Algorithm 2: Keyword Extraction from Crime Data

Input: Sparse graph $G_s = (V, E_s, W_s)$;
Output: Cluster C_K of keywords;
begin
  $C_K = \phi$;
  for *each vertex* $v_i \in V$ *in* G_s **do**
    calculate the total weight using eq. (3);
  end
  Let $v_1, v_2, \cdots, v_n$ be the nodes with max weight in G_s;
  Let g_i be the partition of graph G_s containing v_i and all the incident edges for $i = 1, 2, \cdots, n$;
  Let $G_1 = \{g_1, g_2, g_3, \cdots, g_n\}$;
  $\tilde{G_s} = G_s - G_1$;
  repeat
    Let v_h be the vertex with highest weight and g_h be the partition comprising v_h and all incident edges;
    $\tilde{G_s} = G_s - \{g_h\}$;
    repeat
      for *each partition* $g_i \in G_1$ **do**
        generate a partition $g_i \cup g_h$;
        Calculate the total weight $W(g_i \cup g_h)$;
        Let g_j be the next partition from $\tilde{G_s}$ containing the vertex with highest weight.;
        if $W(g_j \cup g_h) > max\{W(g_j), W(g_h)\}$ **then**
          $\tilde{G_1} = G_1 - \{g_j\}$;
          $g_j = g_j \cup g_h$;
        end
      end
    until *all the vertices and edges are exhausted*;
  until G_s *is a null graph*;
  $\tilde{G_1}$ is the collection of the graph partitions containing the keywords;
  $C_K = C_K \cup \tilde{G_1}$;
  Return C_K;
end

IV. Experimental Results

This section demonstrates the results obtained by the proposed clustering algorithm. This work has been implemented using Python 3.7 with its several modules like numpy, networkx and matplotlib. We chose the online versions of Indian classified newspapers such as 'The Times of India', 'The Hindu', and 'The Indian Express' to collect news reporting on crime against women in Indian states and union territories. After using the suggested graph-based clustering approach, many clusters of keywords are produced. Table IV provides the count of the clusters formed by applying the graph based keyword extraction algorithm.

POS Tags	No. of Clusters
Noun	10
Verb	10
Adjective	6

TABLE IV: Number of clusters formed by proposed keyword clustering

Fig. 2–4 depicts the generated subgraphs containing the keywords for the nouns, verbs and adjectives respectively.

A. Comparative Study

The current work conducted a comparative research by comparing the suggested graph-based algorithm to existing graph-based clustering algorithms available in the literature. The study examined four algorithms: Infomap [13], [14], Louvain [15], Girvan-Newman [16], and Fastgreedy [17]. All

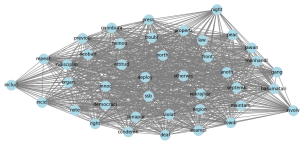

Fig. 2: Subgraphs generated by the proposed keywords extraction algorithm based on nouns

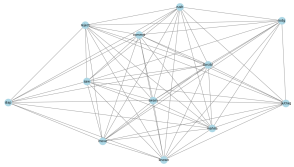

Fig. 3: Subgraphs generated by the proposed keywords extraction algorithm based on verbs

of these methods are graph-based and do not require prior knowledge of the number of clusters. We employed the current crime data for these graph-based clustering approaches, as well as internal cluster evaluation indices, to assess the efficacy of the suggested approach for clustering keywords. Table V provides the number of clusters obtained after using our experimental data on the comparative graph based algorithms.

POS Tag	Fastgreedy	Infomap	Louvain	Girvan-Newman
Noun	17	53	22	4
Verb	20	37	28	17
Adjective	18	40	23	10

TABLE V: Count of clusters generated by the comparative graph algorithms

B. Evaluation Results

Internal cluster assessment indices such as Dunn, Davies-Bouldin, and Silhouette are used to evaluate clusters [18]. The Dunn Index identifies clusters that are confined and well segregated. Thus, it mitigates the intra-cluster distance while increasing the inter-cluster distance. The Dunn index for n clusters is expressed using eq. (4).

$$DN = min_{1\leq a\leq n}\{min_{a\neq b}\frac{d(X_a, X_b)}{max_{1\leq k\leq n}(d(X_k))}\} \quad (4)$$

The intercluster distance in this case between clusters X_a and X_b is shown by $d(X_a, X_b)$. Given n, the number of clusters, and $d(X_k)$, the intracluster distance of cluster X_k. Good clusters are indicated by a higher Dunn index value.

Similarly the Davies-Bouldin or DB index is expressed using eq.(5).

$$DB = \frac{1}{n}\sum_{a=1}^{n} max_{a\neq b}\frac{d(X_a) + d(X_b)}{d(X_a, X_b)} \quad (5)$$

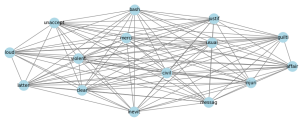

Fig. 4: Subgraphs generated by the proposed keywords extraction algorithm based on adjectives

In this case, n is the number of clusters, while a and b are the cluster labels. Inter-cluster distance $d(X_a, X_b)$ between clusters X_a and X_b is measured as the distance between the cluster centroids. Intra-cluster distances $d(X_a)$ and $d(X_b)$ are the same. Good clustering is shown by the minimum value of the DB index.

To determine the true number of clusters in a data collection, one more really helpful statistic is the Silhouette index of a set of n clusters. This index is calculated for every sample point in every n cluster, and the Silhouette index of the set of n clusters is ultimately the average of all computed values. It is calculated using eq. (6).

$$SI = \frac{1}{C}\sum_{i=1}^{C}\frac{(b_i - a_i)}{max(a_i, b_i)} \quad (6)$$

wherein b_i is the distance from the i-th sample to the closest cluster, C is the number of objects, and a_i is the average distance between the i-th sample and every other sample within its own cluster. A cluster set that is ideal is one that has the highest Silhouette index value. Table VI shows the number of clusters produced using different graph-based methodologies. The results for cluster evaluation is shown in Table VI. The best results obtained for each metric are marked in bold.

Method	Indices		
	DN	DB	SI
KECD	**0.71**	**0.36**	**0.67**
Infomap	0.47	0.36	0.39
Louvain	0.56	0.38	0.48
Fastgreedy	0.41	0.62	0.50
Girvan-Newman	0.45	0.41	0.51

TABLE VI: Evaluation of the proposed approach (KECD) based on internal metrics (in %) on different data

V. Conclusion and Future Work

The current study offers an unsupervised method for obtaining keywords from crime data. Important keywords that can benefit the criminal justice system and the field of criminology are identified using the suggested clustering technique. The keywords stand for the crime terminology, which illustrate various facets of crimes committed against women in India. Through this experimental research project, those facets of crime are revealed. Since nouns, verbs, and adjectives are the best parts of speech tags to use when describing crimes, we

have solely taken them into consideration. However, additional tags can also be used using this strategy. General datasets can also be used using this technique. Through the investigation of other facets of crime pattern analysis, improvisations in the approach will ultimately aid law enforcement agencies in their analysis of criminal activity, thereby providing a comprehensive description of associated activities.

REFERENCES

[1] Z. Xu and J. Zhang, "Extracting keywords from texts based on word frequency and association features," *Procedia Computer Science*, vol. 187, pp. 77–82, 2021.

[2] Y. Zhang, M. Tuo, Q. Yin, L. Qi, X. Wang, and T. Liu, "Keywords extraction with deep neural network model," *Neurocomputing*, vol. 383, pp. 113–121, 2020.

[3] M. Garg, "A survey on different dimensions for graphical keyword extraction techniques: Issues and challenges." *Artificial Intelligence Review*, vol. 54, no. 6, p. 4731–4770, 2021.

[4] N. Firoozeh, A. Nazarenko, F. Alizon, and B. Daille, "Keyword extraction: Issues and methods," *Natural Language Engineering*, vol. 26, no. 3, pp. 259–291, 2020.

[5] S. Pan, Z. Li, and J. Dai, "An improved textrank keywords extraction algorithm," in *Proceedings of the ACM Turing Celebration Conference-China*, 2019, pp. 1–7.

[6] Y. Qian, E. Santus, Z. Jin, J. Guo, and R. Barzilay, "Graphie: A graph-based framework for information extraction," *arXiv preprint arXiv:1810.13083*, 2018.

[7] A. Xiong, D. Liu, H. Tian, Z. Liu, P. Yu, and M. Kadoch, "News keyword extraction algorithm based on semantic clustering and word graph model," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 886–893, 2021.

[8] H. M. M. Hasan, F. Sanyal, and D. Chaki, "A novel approach to extract important keywords from documents applying latent semantic analysis," in *2018 10th International Conference on Knowledge and Smart Technology (KST)*, 2018, pp. 117–122.

[9] L. Zhang, Y. Li, and Q. Li, "A graph-based keyword extraction method for academic literature knowledge graph construction," *Mathematics*, vol. 12, no. 9, 2024.

[10] C. Zhang, H. Wang, Y. Liu, D. Wu, Y. P. Liao, and B. Wang, "Automatic keyword extraction from documents using conditional random fields," *Journal of Computer Information Systems*, vol. 4, pp. 1169–1180, 2008.

[11] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162

[12] P. Das and A. K. Das, "Graph-based clustering of extracted paraphrases for labelling crime reports," *Knowledge-Based Systems*, vol. 179, pp. 55–76, 2019.

[13] M. Rosvall, "Infomap," 2009. [Online]. Available: http://http://www.mapequation.org/code.html

[14] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," 2009, pp. 13–23.

[15] Blondel, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, pp. 1–12, October 2008.

[16] N. M. Girvan, "Community structure in social and biological networks," *Proceedings of National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, June 2002.

[17] Newman, "Modularity and community structure in networks," *Proceedings of National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, May 2006.

[18] P. Das, A. K. Das, J. Nayak, D. Pelusi, and W. Ding, "A graph based clustering approach for relation extraction from crime data," *IEEE Access*, vol. 7, pp. 101 269–101 282, 2019.