

Text similarity algorithms to determine Indian penal code sections for offence report

Ambrish Srivastav, Shaligram Prajapat

Department of Computer Science, International Institute of Professional Studies (IIPS), Devi Ahilya University (DAVV), Indore, India

Article Info

Article history:

Received Mar 25, 2021

Revised Dec 22, 2021

Accepted Dec 29, 2021

Keywords:

Decision support system
Information retrieval system
Law information system
Natural language processing
Text similarity
Vector space model

ABSTRACT

Taking decisions by comparing two text documents is a new innovative idea. Text documents contain details, rules and information related to a domain. The judiciary system is an area where many textual documents are available. In some documents, rules related to the judiciary are mentioned, such as the Indian penal code (IPC) section documents and other documents like first information report (FIR), and Investigation report. contain details of incidents. Our assumption is that the system can help in making the decision by finding the right IPC Section from the result of text similarity between IPC section document and FIR, investigation report. In this research paper, we preface a new research problem to make decisions to suggest appropriate IPC Section for crime related information from user's input by using vector space model and natural language processing techniques.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Ambrish Srivastav

Department of Computer Science, IIPS, DAVV

139, Khandwa Rd, Indrapuri Colony, Indore, Madhya Pradesh (India) 452001

Email: a.srivastav30@gmail.com

1. INTRODUCTION

The decision support system (DSS) is a computerized program used for decision-making activities aimed at growing the business. Presently, due to the progress in the field of computers, all new documents from different areas are being digitalized. Documents related to the judicial system, such as first information reports (FIRs), investigation reports, and judgments are available digitally, in which we can extract any information by implementing a computerized algorithm. In the past decade, some systems were developed to help with decision making by using text similarity algorithms. This system calculates the similarity between two legal documents by using concept based similarity, multi-dimensional similarity [1] and embedding-based methodologies [2]–[4].

Developing DSS to analyze report and finding appropriate Indian penal code (IPC) section according is a new idea. Whenever there is any crime in the society, its information is given to the police and the police are investigate based on that information. The police prepare a comprehensive report (charge sheet) for the court, which mentions sections of the various IPC related to the crime. Knowledge and experience of the sections of the IPC is required to prepare the charge sheet, on the basis of which a correct and appropriate document is prepared for the court. Apart from the police, some other people or organizations can also be users of the system. A lawyer who re-examines the charge sheet and based on his experience prepares the background of the crime and presents it to the offender or victim's side in court. Reading and understanding documents manually such a difficult and time taking task for everyone. If computer program helps in highlighting important information and checking correctness of result according to rules, it will help to understanding document fastly. A common person or organization can also use this

system, with which any crime, deception or violation of rights has taken place. The person or organization has to enter the details of the incident with them in the system.

To use the system, the user will have to enter the information of the incident in the form of natural language text and after analyzing the incident, the system will decide the section of the IPC. Here, we propose a DSS for finding IPC sections (as an appropriate answer) for input of the user. The section of the penal code depends on the various situations, circumstances, some other information of the crime and the definition defined in IPC document. Therefore, analysis of IPC documents and inputs will be necessary. A user may also not write exact word of offense according to penal code document in application, report or query as input then our proposed system finds penal code sections as an appropriate answer and related information for the user. Our idea is to calculate similarity between every sentence of user's input and description of every section of IPC document. According to similarity value, system will suggest list of most appropriate IPC sections for user's input.

In earlier days, DSS was developed for decision making for business purposes, but today, it is evolving for many fields like healthcare, security, medicine, manufacturing, and engineering. In literature, huge work is available for a variety of decision support systems. In recent years there are many various legal/law information systems developed. Quaresma and Rodrigues have proposed a computational linguistic theory (syntactic, semantic analysis and semantic interpretation) based approach to develop a question-answering system for juridical documents in Portuguese language. Query processing by information retrieval and analysis of documents by information extraction are two modules of this question answering systems (QAS). This system contained complete set of decisions from several Portuguese juridical institutions [5]. Tirpude and Alvi have proposed a keyword-based quality assurance (QA) system for legal documents of Indian laws. For this, the author constructs the corpus and knowledge base from legal documents and prepared question dataset with answer type. This system suggested answer of query on the basis of keywords Indexed term dictionary [6]. Kamdi and Agrawal developed question answering system for IPC sections and Indian amendment laws. This QAS select keywords and question type from query and response according answer stored in corpus. Authors define that problem lies on intersection of two domains: Information retrieval (IR) and natural language processing (NLP) [7]. Sangeetha *et al.* have proposed an information retrieval system is designed to retrieve relevant answers about laws. The user query in a system was processed using natural language processing techniques. This system was designed to face dynamic queries from the user end instead of stored question answers [8].

Text processing is an essential part of every natural language based system. Various machine learning approach like decision tree, nearest neighbors, support vector machines, sparse network of windows, naïve bayes and log-linear model (maximum entropy models) experimented for classification of text [8]–[10]. For identifying part-of-speech tagging, name entities and morphological analysis rules-based techniques, Google directory and hidden markov model were developed [11]–[15]. For identifying and removing stop words from text a latent semantic indexing (LSI), SVM-based approach and deterministic finite automata (DFA) were developed [16]–[18]. For solving the issue of statement formation of systematic question Template-based approach proposed. This approach worked on domain-specific Wh-type questions and imperative questions [19].

Calculating text similarity between two different documents is the main task of my research. Various approaches have been proposed by different authors for this work. Mihalcea *et al.* have proposed a corpus-based and knowledge-based measures method of for measuring the semantic similarity of short texts by exploiting the information that can be drawn from the similarity of the component words [20], [21]. Vector space model (VSM) is used for calculating text similarity of small sentences and paragraphs [22]–[25]. Graph-based text similarity (GBTS) algorithm maps Chinese texts into graphs then calculates the similarity of two texts by comparing their graphs [26]. Xue *et al.* presented a method of text similarity computing to the clinical decision support system. Authors improved TF-IDF algorithm and cosine similarity algorithm by combining with eigenvector associated model to determine the case feature weights [27]. Duan and Xu presented short text similarity algorithm for finding similar police incidents. This algorithm was developed from a novel semantic similarity algorithm word mover's distance (WMD) [28]. Jo proposed the version of k-nearest neighbor (KNN) which considers similarity among attributes for computing the similarity between feature vectors [29]. Noufa Alnajran *et al.* proposed heuristic driven pre-processing methodology for enhancing the performance of similarity measures in the context of twitter tweets [30].

2. PROPOSED ARCHITECTURE OF SYSTEM

Based on rationales in previous sections, Figure 1 presents architecture of DSS for finding the most suitable IPC Section of user's input. In the first layer of the system, user input will be analyzed using NLP techniques and in the second layer a knowledge base for the IPC section document will be developed. System consists of several components including-

Text similarity algorithms to determine Indian penal code sections for offence report (Ambrish Srivastav)

- Component for extraction of offence words and crime related information from the user's input query.
- Components for analyzing crime related information and definition of selected IPC sections.
- Relevance matching component for crime: According to the definition of particular IPC sections.
- Get and show most appropriate IPC sections.

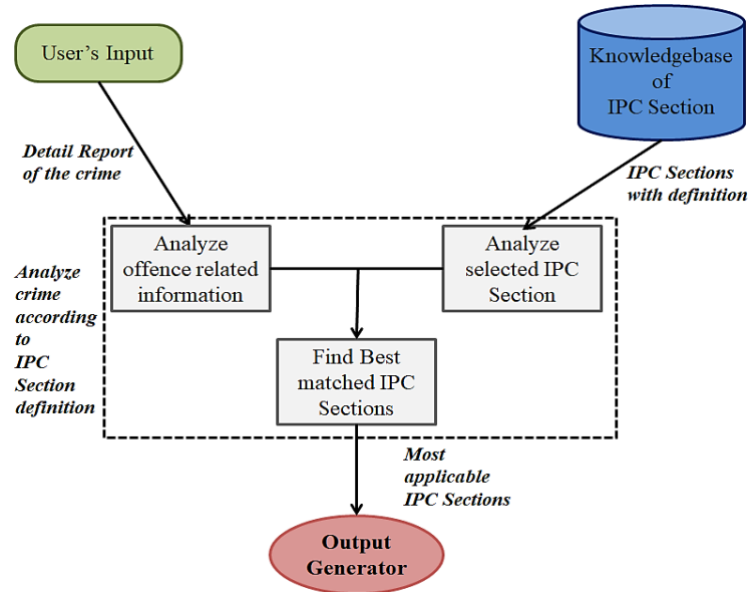


Figure 1. Proposed architecture of system

3. METHOD

IPC document and offence report are two different type of unstructured text. Development of such a system for determines most appropriate IPC Sections for a crime report from unstructured text document of IPC is difficult task. We identify the following steps to achieve our goal.

- Step 1: Developing a corpus for IPC section document. The IPC document distributes 511 sections in 23 chapters. Each chapter describes some kind of crime and conditions. In a corpus of IPC section we include four parts (IPC section no, root, offence and description of section).
- Step 2: Apply method of calculating the text similarity between input text and description of IPC section. Semantic similarity is a measure of conceptual distance between two objects, based on the correspondence of their meanings [31].

The IPC section description text and user input text are two different types of documents and there is very little chance that they are lexical similar. Our objective is to calculate semantic similarity between pair of every sentence of selected IPC section description text with every sentence of user's input. To calculate similarity, follow the following steps:

- i) Apply pre-processing in IPC Section description text and user's input text. We used natural language processing toolkit, NLTK for implementing pre-processing. Steps are:
 - Tokenization: Tokenization is a procedure of splitting a sentence into list of words.
 - Lower casing: Convert all words in common case (most preferable lower case) because in NLP same word in different case treated as a different word.
 - Stop words removal: In a text document, there are so many words (like 'is', 'was', 'a', and 'the'.) that do not signify any importance in processing. So, these words must remove from document before processing.
 - Stemming/lemmatization: Stemming and lemmatization is a process of transforming a word to its root form. Lemmatization works better then stemming for converting a word to its root form.
 - After cleaning text document, we found most important words in IPC section description and user's input for further processing.
- ii) Use filtered IPC Section description words as a term. Apply feature engineering for finding feature of user's input text as a vector from term So, feature engineering technique will calculate vector value

according to presence of terms or its synonyms word in user’s input. There are several techniques that apply to derive relevant features from a text document.

3.1. Vector space model

Vector space model is a matrix representation of list of documents and corpus of words. Every row represents individual document and columns represent words of corpus. Cell store value ‘0’ or ‘1’. ‘0’ means that word not present in document and ‘1’ indicates word occurred in document. In our problem vector matrix shows occurrence of terms (selected feature of particular IPC section) in a text document (user’s input) and according to cells value we can calculate appearance of IPC Section in sentence. In the user’s input, there may be many sentences that are not related to the IPC section. If the vector value of all the words in the sentence is ‘0’ then system will ignore that sentence for score calculation. We create vectors for description of each IPC section and every paragraph of user’s input and the system will use these vectors for further calculations. There are some tools for converting text document into a vector.

i) **CountVectorizer:** CountVectorizer is a tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text. Let consider the example for some filtered IPC Section description:

- D0: public nuisance illegal omission cause common injury danger
- D1: unlawfully negligent act likely spread infection disease dangerous life
- D2: malignant act likely spread infection disease dangerous life

Sample result of CountVectorizer shows frequency of words in each document (D0, D1 and D2) in Table 1. If word appears in document then frequency of word becomes ‘1’ otherwise it will be ‘0’.

ii) **TF-IDF:** TF-IDF stands for term frequency-inverse document frequency. In this model, we take term frequency and inverse document frequency as parameters to decrease the weight of the terms appearing commonly in all the sentences. Formulas of calculating TF-IDF stepwise are:

- $tf(t, d) = \text{count of } t \text{ in } d / \text{number of words in } d$ //term frequency
- $df(t) = \text{occurrence of } t \text{ in documents}$ //document frequency
- $idf(t) = \log(N/df(t))$ //inverse document frequency
- $tf-idf(t, d) = tf(t, d) * idf(t)$

Sample result of TF-IDF shows frequency of words in each document (D0, D1 and D2) in Table 2. Frequency of each word calculated by its appearance in particular document and all documents.

Table 1. Sample IPC section vector using CountVectorizer

	act	cause	common	danger	public	spread	Unlawfully
0	1	0	0	0	0	1	1
1	1	0	0	0	0	1	0
2	0	1	1	1	1	0	0

Table 2. Sample IPC section vector using TF-IDF

	act	cause	common	danger	public	spread	Unlawfully
0	0.309228	0	0	0	0	0.309228	0.406598
1	0.33847	0	0	0	0	0.33847	0
2	0	0.353553	0.353553	0.353553	0.353553	0	0

- **Step 3:** Calculate Cosine similarity between vectors of every paragraph of users input with vector of each IPC Section description. Cosine similarity measures the similarity between two vectors of an inner product space as shown in Figure 2. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis. Values range between -1 and 1, where -1 is perfectly dissimilar and 1 is perfectly similar.

$$\text{Similarity (A, B)} = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

- **Step 4:** According to this calculation of cosine similarity, system will show list of most appropriate IPC sections that’s closely related to users input. Here one document is description of IPC section and another document is paragraph of user’s input.

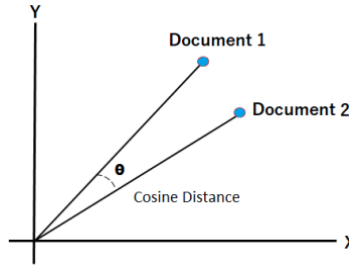


Figure 2. Cosine distance similarity

4. RESULTS AND DISCUSSION

4.1. Development of corpus

There are 511 sections in IPC document that are divided into 23 chapters. We have selected 4 chapters of the IPC document, which are chapters 14, 15, 16 and 22, to prove the presumed correctness of our proposed work. We developed corpus for sections (around 120) of these chapter as shown in Table 3.

4.2. Select complain for input

We have selected the complaint text as shown in Figure 3 related to these chapters as the input query. These complaints are available in the form of FIR. on the official portal of state police in India. The FIR is divided into paragraphs which contain the offense and its related information.

4.3. Similarity calculation

Count vector and TF-IDF model applied to calculate text similarity between each paragraph of complaint with description of each section and found list of most appropriate ‘10’ IPC sections that’s most related to complain as shown in Table 4. As a result both models produce some list of IPC sections. This list and its sequence are different in result of both model but most of sections are common related to complain. Based on the output of these models, the system can act as decision support for the user.

Table 3. Corpus for IPC section document

Section	Root	Offence	Description
268	nuisance	Public nuisance	Public nuisance, illegal omission which causes any common injury, danger
269	negligently	Negligent act	Unlawfully, Negligent act likely to spread infection of disease dangerous to life
270	malignant	Malignant act	Malignant act likely to spread infection of disease dangerous to lifedangerous to life

Jayendra Saraswathi is the head of the Kanchi Mutt, one of a prominent Hindu monastic institutions in the country. An investigative journalist named Dhanasekaran Prakash in the Tamil weekly Nakkeeran alleged the reasons of the murder being the continuous infuriation by Sankararaman against Jayendrar and Kanchi Mutt.]

Figure 3. Sample complaint text

Table 4. Comparison of count vector and TF-IDF result

Count Vector Result	TF-IDF Result
related_ipcs_index [118 42 48 41 49 51 66 26 123 43]	related_ipcs_index [118 42 66 26 48 123 49 41 141]
('IPC', 364, ':', 'Kidnapping or abducting in order to murder')	('IPC', 364, ':', 'Kidnapping or abducting in order to murder')
('IPC', 303, ':', 'Punishment for murder by life-convict')	('IPC', 303, ':', 'Punishment for murder by life-convict')
('IPC', 307, ':', 'Attempt to murder')	('IPC', '320F', ':', 'Grievous hurt')
('IPC', 302, ':', 'Punishment for murder')	('IPC', 290, ':', 'Punishment for public nuisance in cases not otherwise provided for')
('IPC', 308, ':', 'Attempt to commit culpable homicide')	('IPC', 307, ':', 'Attempt to murder')
('IPC', 310, ':', 'Thug')	('IPC', '366B', ':', 'Importation of girl from foreign country')
('IPC', '320F', ':', 'Grievous hurt')	('IPC', 308, ':', 'Attempt to commit culpable homicide')
('IPC', 290, ':', 'Punishment for public nuisance in cases not otherwise provided for')	('IPC', 302, ':', 'Punishment for murder')
('IPC', '366B', ':', 'Importation of girl from foreign country')	('IPC', '376C', ':', 'Intercourse by superintendent of jail and remand home')
('IPC', 304, ':', 'Punishment for culpable homicide not amounting to murder')	('IPC', 310, ':', 'Thug')

5. CONCLUSION

This research paper starts with an introduction of a problem in judicial system and finds solution by using decision support system (DSS). DSS aims to help make the best decision based on existing information. Over the past few decades, a number of information retrieval (IR) system and question answering systems (QAS) have been developed to find result and answers in a limited specific area. IR system and QAS takes single line question and apply NLP techniques to extract keyword and search result. Here we propose the architecture of DSS for crime incident documents which suggest the list of most applicable IPC section by comparing the user input document and IPC section document by vector space model. Our proposed system enhances the working of typical question answering system and help to take decision on the basis of result. In the future, some other text similarity algorithms such as word2vec, doc2vec, and BERT (sentence transform). will use to check the accuracy of the system.

ACKNOWLEDGEMENT

I want to thank my supervisor Dr. Shaligram Prajapat, Associate Professor in IIPS DAVV, Indore not only for his continued support but for the motivation and fruitful advises in accomplishing this task.





REFERENCES

- [1] R. S. Wagh and D. Anand, "Legal document similarity: a multi-criteria decision-making perspective," *PeerJ Computer Science*, vol. 6, Art. no. e262, Mar. 2020, doi: 10.7717/peerj-cs.262.
- [2] A. Mandal, R. Chaki, S. Saha, K. Ghosh, A. Pal, and S. Ghosh, "Measuring similarity among legal court case documents," in *Proceedings of the 10th Annual ACM India Compute Conference on ZZZ-Compute '17*, 2017, pp. 1–9, doi: 10.1145/3140107.3140119.
- [3] P. Bhattacharya, K. Ghosh, A. Pal, and S. Ghosh, "Methods for computing legal document similarity: a comparative study," *Computer Science*, Apr. 2020.
- [4] S. Renjit and S. M. Idicula, "Similarity in legal texts using document level embeddings," *CUSAT NLP@AILA-FIRE2019*, pp. 25–30, 2019.
- [5] P. Quaresma and I. P. Rodrigues, "A question answer system for legal information retrieval," in *Proceedings of the 2005 conference on Legal Knowledge and Information Systems: JURIX 2005: The Eighteenth Annual Conference*, 2005, pp. 91–100.
- [6] S. C. Tirdude and D. A. S. Alvi, "Closed domain keyword based question answering system for legal documents of IPC sections Indian laws," *International Journal of Innovative Research in Computer and Communication Engineering*, 2015.
- [7] R. P. Kamdi and A. J. Agrawal, "Keywords based closed domain question answering system for Indian penal code sections and Indian amendment laws," *International Journal of Intelligent Systems and Applications*, vol. 7, no. 12, pp. 57–67, Nov. 2015, doi: 10.5815/ijisa.2015.12.06.
- [8] D. Sangeetha, R. Kavyashri, S. Swetha, and S. Vignesh, "Information retrieval system for laws," in *2016 Eighth International Conference on Advanced Computing (ICoAC)*, Jan. 2017, pp. 212–217, doi: 10.1109/ICoAC.2017.7951772.
- [9] D. Zhang and W. S. Lee, "Question classification using support vector machines," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval-SIGIR '03*, Aug. 2003, p. 26, doi: 10.1145/860435.860443.
- [10] P. Blunsom, K. Kocik, and J. R. Curran, "Question classification with log-linear models," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval-SIGIR '06*, 2006, p. 615, doi: 10.1145/1148170.1148282.
- [11] J. Liu and L. Birnbaum, "Measuring semantic similarity between named entities by searching the web directory."
- [12] R. Ageishi and T. Miura, "Named entity recognition based on a Hidden Markov Model in part-of-speech tagging," in *2008 First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)*, Aug. 2008, pp. 397–402, doi: 10.1109/ICADIWT.2008.4664380.
- [13] Zhang Youzhi, "Research and implementation of part-of-speech tagging based on Hidden Markov Model," in *2009 Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA)*, Nov. 2009, pp. 26–29, doi: 10.1109/PACIIA.2009.5406648.
- [14] R. Cretulescu, A. David, D. Morariu, and L. Vintan, "Part of speech tagging with Naïve Bayes methods," in *2014 18th International Conference on System Theory, Control and Computing (ICSTCC)*, Oct. 2014, pp. 446–451, doi: 10.1109/ICSTCC.2014.6982457.
- [15] S. P. Singh, A. Kumar, and H. Darbari, "Deep neural based name entity recognizer and classifier for English language," in *2017 International Conference on Circuits, Controls, and Communications (CCUBE)*, Dec. 2017, pp. 242–246, doi: 10.1109/CCUBE.2017.8394152.
- [16] A. N. K. Zaman, P. Matsakis, and C. Brown, "Evaluation of stop word lists in text retrieval using latent semantic indexing," in *2011 Sixth International Conference on Digital Information Management*, Sep. 2011, pp. 133–136, doi: 10.1109/ICDIM.2011.6093315.
- [17] S. Xu, G. Cheng, and F. Kong, "Research on question classification for automatic question answering," in *2016 International Conference on Asian Language Processing (IALP)*, Nov. 2016, pp. 218–221, doi: 10.1109/IALP.2016.7875972.
- [18] S. Behera, "Implementation of a finite state automaton to recognize and remove stop words in english text on its retrieval," in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, May 2018, pp. 476–480, doi: 10.1109/ICOEI.2018.8553828.
- [19] K. Pawar and U. Shrawankar, "Question systematization using templates," *3rd International Conference on Computing for Sustainable Global Development*, 2016.
- [20] R. Mihalcea and C. C. C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *{AAAI}'06: {Proceedings} of the 21st national conference on {Artificial} intelligence*, Jul. 2006, vol. 1, pp. 775–780.
- [21] W. H. Gomaa and A. A. Fahmy, "A survey of text similarity approaches," *International Journal of Computer Applications*, vol. 68, no. 13, pp. 13–18, Apr. 2013, doi: 10.5120/11638-7118.





- [22] H. Dong, J. Wu, X. Zhao, and Y. Li, "Study on the calculation of text similarity based on key-sentence," in *2010 International Conference on E-Business and E-Government*, May 2010, pp. 1952–1955, doi: 10.1109/ICEE.2010.493.
- [23] W. Yih, K. Toutanova, J. C. Platt, and C. Meek, "Learning discriminative projections for text similarity measures," in *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, 2011, pp. 247–256.
- [24] P. Shrestha, "Corpus-based methods for short text similarity," in *TALN 2011*, 2011, pp. 1–6.
- [25] G. Liu and H. Wang, "A recursive descent evaluation algorithm on policy context similarity," in *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, May 2018, pp. 21–25, doi: 10.1109/ICAIBD.2018.8396160.
- [26] Z. Liu and X. Chen, "Mapping texts into graphs: An improved text similarity algorithm," in *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*, Dec. 2012, pp. 1357–1361, doi: 10.1109/ICCSNT.2012.6526173.
- [27] T. Xue, Y. Yuan, Q. Fu, H. Gu, S. Zhang, and C. Wang, "The application of text similarity computing in the clinical decision support system," Nov. 2014, doi: 10.1109/ccis.2014.7175759.
- [28] L. Duan and T. Xu, "A short text similarity algorithm for finding similar police 110 incidents," in *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, Nov. 2016, pp. 260–264, doi: 10.1109/CCBD.2016.058.
- [29] T. Jo, "Using k-nearest neighbors for text segmentation with feature similarity," in *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, Jan. 2017, pp. 1–5, doi: 10.1109/ICCCCEE.2017.7866706.
- [30] N. Alnajran, K. Crockett, D. McLean, and A. Latham, "A heuristic based pre-processing methodology for short text similarity measures in microblogs," in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 1627–1633, doi: 10.1109/HPCC/SmartCity/DSS.2018.00265.
- [31] D. Lin, "An information-theoretic definition of similarity," in *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 296–304.

BIOGRAPHIES OF AUTHORS



Amrish Srivastav     is a research scholar at, Devi Ahilya University (DAVV), Indore and having teaching experience approx 10 years in field of Computer Science and Engineering. He graduated in 2009 from I.E.T.E, New Delhi and received his Master's degree in 2011 from I.E.T. DAVV. His research interests are Artificial Intelligence, Natural Language Processing and Machine Learning. He can be contacted at email: a.srivastav30@gmail.com.



Dr. Shaligram Prajapat     has been working in academics as an educationist, teacher, researcher and learner since past 2 decades. He has executed many academic and research projects as a part of Devi Ahilya university, India. In research career, with Ph.D. from Maulana Azad National Institute of Technology (M.A.N.I.T.) in Computer Applications from Bhopal India and Master of Philosophy (Computer Science) from Devi Ahilya University Indore, he has more many research publications in International journals listed in web of science and Scopus. He can be contacted at email: shaligram.prajapat@gmail.com.