Case Finder: Criminal Case Records Similarity Prediction Using NLP

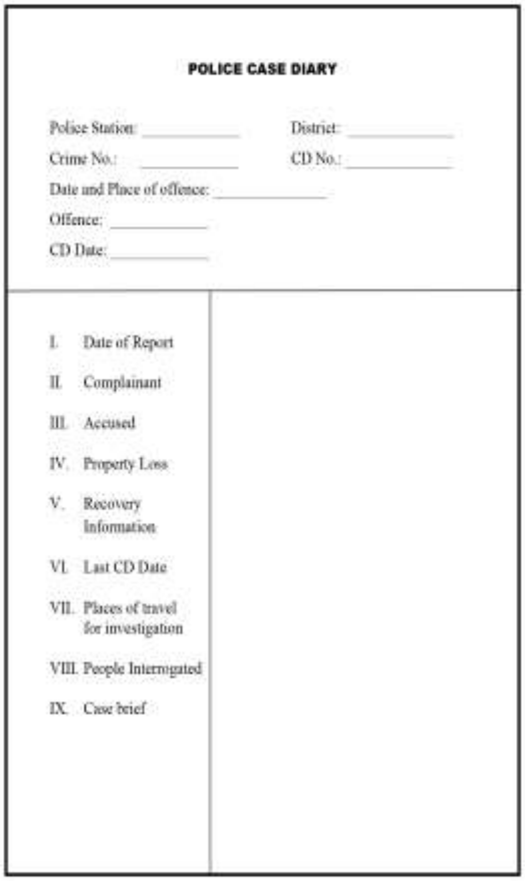
Ann Jerin Sundar[1], Shine J S[2], Tharun P Karun[3]
[1, 3]Dept of Computer Application
[2] Assistant Professor, Dept of English
[1, 3] College of Engineering, Trivandrum, Kerala, India
[2]Karunya deemed-to-be-University, Coimbatore,Tamil Nadu, India

Abstract- *A case diary encompasses the crucial details of a registered offence.Often known as the police diary, it encloses explicit details pertaining to the accused, such as the criminal background history, behavioural patterns, medical reports pointing out psychological complications and other particulars. Referring to previous case diaries of similar criminal background will ease the investigation process largely. Recognition of crimes having similar patterns of execution can eliminate the decelerating factors, resulting in a more efficient investigation process. It can also identify behavioural similarity in cases, revealing a serial offence with a single offender. This study proposes and deals with a data analytical methodology that is efficacious in culminating a crime investigation process more conveniently. The efficacy is achieved with the identification of case records, similar to that of the newly input case, through the incorporation of Natural Language Processing techniques.*

Keywords- Natural Language Processing, gensim, similarity retrieval, python, criminal case record.

I. INTRODUCTION

Criminal investigation is an applied science that involves the study of facts that are then used to inform criminal trials. A complete criminal investigation can include searching, interviews, interrogations, evidence collection and preservation and various methods of investigation. A Case Diary, often known as a Police Diary is a crucial element in a crime investigation process. It comprises of crucial elements dating from the time of registering a case to the completion of it's investigation process. The head Police investigator in charge, records and appends data regarding the investigation of the ongoing case into a Case Diary on a daily basis, and it remains extremely confidential. This document serves as a key aid in investigating cases more systematically and conveniently. It will assist as a reference aid for the many Police officials appointed in-charge of the same case in the event of longer investigation processes ranging to years or decades.



POLICE CASE DIARY

Police Station: ____________ District: ____________
Crime No.: ____________ CD No.: ____________
Date and Place of offence: ____________
Offence: ____________
CD Date: ____________

I. Date of Report
II. Complainant
III. Accused
IV. Property Loss
V. Recovery Information
VI. Last CD Date
VII. Places of travel for investigation
VIII. People Interrogated
IX. Case brief

Fig 1. Police Case Diary Sample Format.

Case Finder relieves the investigating official of searching for similar kind of cases that might help in furthering the investigation. It is developed with the sole aim of generating a list of case records from the criminal database, that could be correlated in terms of patterns or behaviorism of carrying out a particular crime. This can shed light on serial offences that could lead to a serial offender or prime suspects in cases which got dismissed due to insufficient resources or evidences.

The paper is categorized into four sections with Introduction being the first part. The second section deals with the Proposed Method. The third highlights the System Design and finally section four concludes the paper.

II. PROPOSED METHOD

This study proposes a data analytical methodology that will ease the work and effort that is to be put in by an investigating Police officer. Case Finder comprises of three modules:

Dataset Processing
Query Document Processing
Tkinter User Interface

Case Finder is initiated with inputting a new criminal case record's Case Brief Document into the system. It is thus expected to enlist the case records that share similar patterns of crime behaviourism or modus of carrying out the offence as that of the input record, via a Tkinter interface. The main python library incorporated for this is the gensim library[1] , that proved to be of great efficiency for Natural Language Processing and similarity retrieval. Other significant tools included are the nltk[3], pandas[4], and Tkinter[5] libraries

III. SYSTEM DESIGN

The overall design of the system is based on the concept of Natural Language Processing and Document Similarity.

A. NLP SIMILAR CASE PREDICTION

The comparison between different case diaries is performed by assessing the similarity between the different documents in the criminal record database. This process outputs the required documents that follow a similar pattern of criminal history and investigation.

The initial step of document preprocessing is performed for the purpose of noise reduction that includes removal of stopwords, stemming and furthermore. Stopwords are those terms which are given potent roles in the structuring of a grammatically accurate sentence, but may be omitted in the mining process for a more efficient output. They include commonly used prepositions, conjunctions, articles etc. Stemming refers to the reduction of inflectional forms to their root form. This technique helps in retrieving all the possible values for a required query devoid of clarity, thereby reducing ambiguity. The required field in the case diary that depicts the actual patterns of the case investigation is input for preprocessing.

Similarity measurement among the input documents is carried out in several steps. The gensim library tools are the main tools incorporated for this purpose. The important metric processing is carried out through Vectorization and Distance computation.

1) Vectorization :

Vectorization is the transformation of pre processed documents into a vector of numbers. Initially, it formulates a set of words which are common in the input documents. Subsequent vectorization is performed on these words through various methods. The most preferred method is the TF-IDF method.

TF-IDF refers to Term Frequency - Inverse Document Frequency, which scores the importance of the set of words on the basis of their frequent appearance in the document. This score or weight of a particular term is a statistical measure which gets ranked higher in proportion with its frequent appearance. Greater the score, greater is the importance of that term. The tf-idf weight is computed by calculating the respective TF (Term Frequency) and Inverse Document Frequency (IDF).

Term Frequency measures the frequency of the occurrence of a particular term in the input criminal case record, and it is document specific. The longer the record, more is the probability of the term's occurrence in it. Hence, the document length plays an inevitable role in the computation of TF. Consider the word 'assault'.

TF(assault) = (Number of times 'assault' appears in a record) / (Total number of terms in the record)

Inverse Document Frequency calculates how significant the term is across multiple records. A term is not considered as a unique identifier if it is frequent across multiple records, thereby weighing it down and providing a lower score. Consider the word 'is' which can be omitted in the mining process.

IDF(is) = log_e (Total number of records / Number of records with 'is' in it)

2) Distance Computation :

Cosine similarity is the similarity metric used in the computation of distance between two non zero vectors. It is computed by taking the dot product of two given vectors.

The resulting metric based on the average weightage given to the respective terms in a document, will assist in the retrieval of a predefined number of ranked documents that depict similar patterns of criminal background and investigation approach.

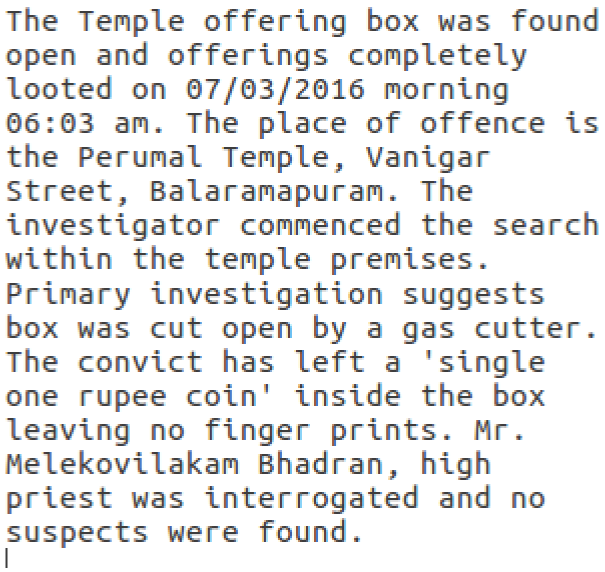
Let the two vectors be A and B, then the cosine similarity shall be computed by,

$$\cos\theta = \frac{A.B}{|A||B|} = \frac{\sum_{i=1}^{n} A_iB_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} ,$$

where A_i and B_i are components of vector A and B respectively.

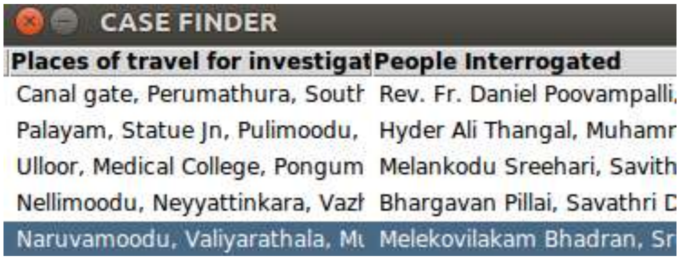
IV. EXPERIMENTAL RESULTS

The initial version of the Case Finder system showed satisfactory results. With the document input of the case brief document, the system predicted the similar case records with all the associated attributes from the criminal case records database.



```
The Temple offering box was found
open and offerings completely
looted on 07/03/2016 morning
06:03 am. The place of offence is
the Perumal Temple, Vanigar
Street, Balaramapuram. The
investigator commenced the search
within the temple premises.
Primary investigation suggests
box was cut open by a gas cutter.
The convict has left a 'single
one rupee coin' inside the box
leaving no finger prints. Mr.
Melekovilakam Bhadran, high
priest was interrogated and no
suspects were found.
```

Fig 2. Sample Case Brief test document.



Places of travel for investigat	People Interrogated
Canal gate, Perumathura, Soutl	Rev. Fr. Daniel Poovampalli,
Palayam, Statue Jn, Pulimoodu,	Hyder Ali Thangal, Muhamr
Ulloor, Medical College, Pongum	Melankodu Sreehari, Savith
Nellimoodu, Neyyattinkara, Vazl	Bhargavan Pillai, Savathri C
Naruvamoodu, Valiyarathala, Mu	Melekovilakam Bhadran, Sr

V. CONCLUSION & FUTURE SCOPE

The Case Finder system is developed keeping in mind the working procedures of a criminal investigation. For this the researcher consulted several officials of the Kerala Police Department[2] and inference was drawn from them. The purpose of the Case Finder system is to list out case records that share similar patterns or behaviourism in carrying out an offence, as to that of an input case record. In the current scenario such a system is non-existent, making the user or the investigator check for outdated records manually, thereby making the process tedious and time consuming. The user is made aware of the previous cases investigated by several officers thereby assisting in the investigation process for deriving useful patterns of modus pointing out a serial offence.

VI. ACKNOWLEDGMENT

We are gratified to the Department of Computer Application, College of Engineering, Trivandrum and each & everyone who helped us in carrying out this research work.

REFERENCES

[1] https://pypi.org/project/gensim/

[2] https://Keralapolice.gov.in/

[3] https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial/

[4] https://pypi.org/project/pandas/

[5] https://docs.python.org/2/library/tkinter.html