JURNAL LOCUS: Penelitian & Pengabdian
Volume 5 No. 4 April 2026
E-ISSN 2829-7334| P-ISSN 2829-5439
Hompage: https://locus.rivierapublishing.id/index.php/jl

Automated Entity Extraction for Building Crime Anatomy from Preliminary Investigation Reports

Febrianto[1]*, Ade Romadhony[2], Yudi Prayudi[3]
Universitas Telkom, Indonesia[1, 2, 3]
Email: febryblctelkom@student.telkomuniversity.ac.id[1]*,
aderomadhony@telkomuniversity.ac.id[2], yprayudi@telkomuniversity.ac.id[3]

Keywords:
Entity Extraction, Crime Anatomy, Preliminary Investigation Reports, Named Entity Recognition (NER), BERT-based Model

Abstract

The increasing volume and complexity of legal documents in criminal investigations pose significant challenges for investigators, particularly due to reliance on manual document review processes that are time-consuming and prone to human error. This study aims to develop an automated system for extracting and structuring key information from preliminary investigation reports to support efficient case analysis. The research employs a Natural Language Processing (NLP) approach using Named Entity Recognition (NER) based on transformer models such as BERT, combined with rule-based Regular Expressions (Regex) to construct a structured "crime anatomy." The methodology involves data collection from Indonesian police reports, manual annotation using the IOB tagging scheme, model training and evaluation using precision, recall, and F1-score, and implementation through a multi-stage processing pipeline. The findings indicate that the integration of NER and Regex significantly improves entity extraction accuracy and structural consistency, with hybrid models achieving near-perfect performance compared to standalone NER models. The system successfully transforms unstructured legal narratives into structured summaries containing key elements such as actors, time, location, modus operandi, and evidence. In conclusion, the proposed approach enhances the efficiency, accuracy, and reliability of criminal case analysis, offering a scalable solution for modern law enforcement and contributing to improved decision-making processes.

INTRODUCTION

Currently, manual document review makes it difficult for investigators to analyze legal documents because they must read each report individually (Bernardi et al., 2023; Klarin, 2024; Marzi et al., 2025; Surahman & Wang, 2022; Taquette & Borges da Matta Souza, 2022). Each document is typically lengthy, uses complex legal language, and contains numerous details about chronology, parties involved, and descriptions of evidence, all of which demand significant time and concentration to fully understand. In practice, this workload is further compounded by the fact that investigators often handle multiple cases in parallel, making it challenging to read, examine, and compare all documents in depth within limited time (Bjelobaba et al., 2025; Halford, 2026; Tan et al., 2022; Walter et al., 2025).

Reliance on manual reading not only slows down the document review process but also increases the risk of various human errors (Akerele et al., 2024; Alqahtani et al., 2023;

Morais et al., 2022; Salunkhe et al., 2025). Investigators may overlook important information, misinterpret relationships between events, or fail to notice inconsistencies across reports when they must process many documents within a short period. At the same time, the need to summarize the essence of a case, understand the chronology, and identify the involved parties quickly and accurately is increasingly critical, especially in the early case conference stage, where strategic decisions must be made based on the available information.

As the volume and complexity of investigation documents grow, manual approaches become increasingly inefficient and difficult to maintain as the sole method for understanding the full content of case files (Alevizos, 2025; Allioui & Mourdi, 2023; Balcıoğlu et al., 2024; Mahadevkar et al., 2024; Salunkhe et al., 2025). This situation creates a need for technology capable of automatically capturing the gist of a collection of documents so that key information from all reports can be presented in a more concise, structured, and easily analyzable form (Černý et al., 2026; Feldhoff et al., 2025; Shanthi, 2025). Such technology is expected to reduce the cognitive burden on investigators, accelerate the review process, and ultimately improve the quality of decision-making in criminal case handling.

In this context, the present thesis proposes the use of Natural Language Processing (NLP) techniques based on Named Entity Recognition (NER) to extract important entity information from all investigation documents related to a case. NER is employed to identify and classify core legal entities such as the complainant, victim, suspect, witness, tempus delicti, locus delicti, modus operandi, and evidence, which are scattered across various reports and investigation documents. In this way, information that was originally dispersed and unstructured can be transformed into a set of more organized entities that are ready for subsequent processing.

Once these entities have been extracted, this study utilizes regular expression (regex) techniques to arrange and formalize the identified information into a structure referred to as the anatomy of crime. Regex is used to detect specific textual patterns, such as date formats, case numbers, descriptions of modus operandi, and standardized expressions of evidence, so that each entity can be placed consistently in its appropriate position within the crime anatomy structure. This combined NER–regex approach is expected to produce a case representation that is concise yet explicitly preserves the essential elements of the criminal act and is easy to interpret.

The resulting anatomy of crime takes the form of a structured summary that consolidates the parties involved, the time and place of the incident, a brief chronology, modus operandi, and the main pieces of evidence into a single, systematic view. This structured representation is designed to be directly usable at the initial stage of the case conference, when investigators and decision-makers require a comprehensive overview of a case within a relatively short time. With the anatomy of crime, the case presentation process no longer needs to rely solely on rereading all investigation documents but can begin with a directed summary automatically generated by the system.

The research focuses on developing a BERT-based NER model that is integrated with regex rules to construct an anatomy of crime from collections of investigation documents. The resulting system is expected to provide an initial solution to the key problems faced by investigators, namely the difficulty of understanding legal documents manually and the lengthy time required to distill the core of a case. Through this approach, the study aims to

establish a technological foundation that not only accelerates and improves the accuracy of case analysis but also contributes to more efficient, transparent, and accountable criminal case management.

Police officers play a crucial role in managing various criminal incidents and administrative matters, including the preparation of criminal case reports, traffic accident reports, and death reports that function as essential legal documents for law enforcement and supporting institutions such as insurance companies. Because these documents may be used at the Supreme Court level, they must be accurate, clear, comprehensive, and timely. However, globally, the manual process of collecting and processing information in such documents often leads to delays and is highly prone to human error, including inaccurate data entry, incomplete information, duplication, delayed submission, and data loss.

International studies report that approximately 35% of errors originate from incorrect data input, 30% from incomplete reports, 15% from document duplication, and 20% from delayed or unretrievable reports, illustrating the vulnerability of manual reporting systems. These inefficiencies not only hinder the speed and accuracy of legal processes but also impact the quality of evidence-based decision-making. As a result, there is an urgent need for reliable technological solutions that can streamline information extraction, reduce human dependency, and ensure the integrity of police documentation.

At the national level, particularly in Indonesia, the need for digitalization and automation of investigative reporting is increasingly important due to high case volumes and limited resources. Advances in Artificial Intelligence (AI), especially NLP-based technologies such as Named Entity Recognition (NER) and BERT-based summarization, offer opportunities to accelerate information processing, enhance data accuracy, support data-driven decision-making, and reduce error rates. Therefore, the adoption of automated investigation document processing systems is seen as a highly relevant and strategic solution to address fundamental challenges in the current manual reporting framework.

METHOD

The research adopted a systematically structured methodology to address the growing complexity of legal documents and the need for automated, accurate, and efficient investigation support systems. It began with problem identification, requirements analysis, and a literature review, which revealed the limitations of manual document processing and the performance degradation of general NER models when applied to legal texts. These findings led to the formulation of a domain-specific research design incorporating specialized datasets, advanced transformer-based NLP models, and rigorous evaluation to ensure methodological validity and scientific contribution.

The development phase focused on constructing an automated investigation report system that integrated transformer-based NER models such as BERT, IndoBERT, RoBERTa, and XLM-RoBERTa, enhanced with spaCy pipelines and annotated Indonesian legal datasets. Manual annotation using the IOB tagging scheme was conducted to create reliable ground-truth data, capturing key entities such as persons, locations, dates, legal articles, and evidence. The system employed a multi-stage pipeline in which machine learning–based entity extraction was followed by a rule-based regular expression (regex) structuring layer

that assigned legal meaning to entities, converting linguistic outputs into structured crime anatomy components.

Implementation was conducted using real Indonesian National Police investigation reports following standardized SOP formats. Documents were processed through tokenization, normalization, NER inference, regex-based structuring, and summarization modules, with performance evaluated using precision, recall, F1-score, and ROUGE metrics against manually annotated benchmarks and expert assessments. The results demonstrated strong entity detection capability and showed that the NER + regex framework significantly improved structural consistency and legal interpretability, enabling unstructured narratives to be transformed into accurate, scalable, and legally meaningful investigation report models.

RESULT AND DISCUSSION

A. Google Colab Preparation

Google Colab is a free cloud-based service that allows users to write and run Python code in their browser without installation, much like an interactive Jupyter notebook. This service is particularly useful for machine learning, data analysis, and education because it provides free access to computing resources like GPUs and TPUs.

To run the AnatomiCrime Extractor program without manual installation on a local computer, users can utilize a pre-installed notebook in Google Colab. This notebook contains all the source code, model configuration, and an example of the execution flow from document loading to creating an anatomy of a crime. Users only need a Google account and an internet connection. Users can access the AnatomiCrime Extractor program notebook through the following link: https://colab.research.google.com/drive/1DscVSGZyEbNzGcO0PTbhkxKoXSdAgN_i?usp=sharing

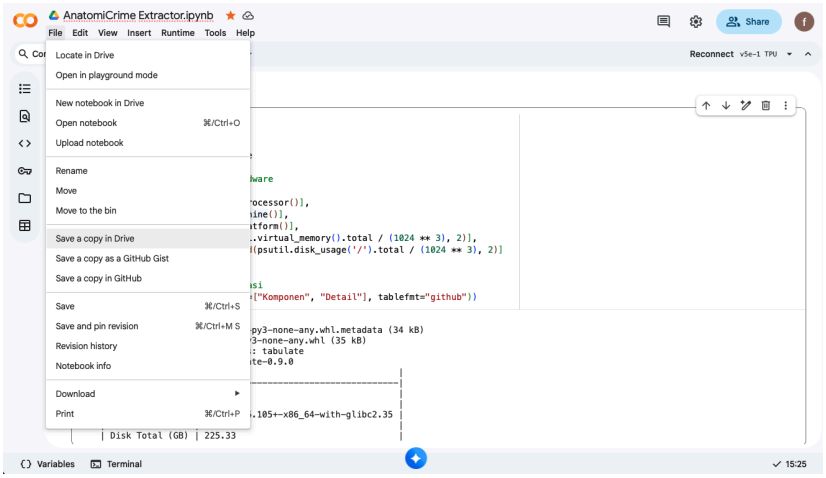


Figure 1. Program AnatomiCrime Extractor

B. Preparation of Police Report Documents

Figure 2. SOP for Criminal Investigation Administration

The documents used come from the Indonesian National Police in the form of an Indonesian language criminal investigation or inquiry report that follows the official administrative format in accordance with the Indonesian National Police Criminal Investigation Agency Regulation Number 1 of 2022 concerning Standard Operating Procedures for Criminal Investigation Administration.

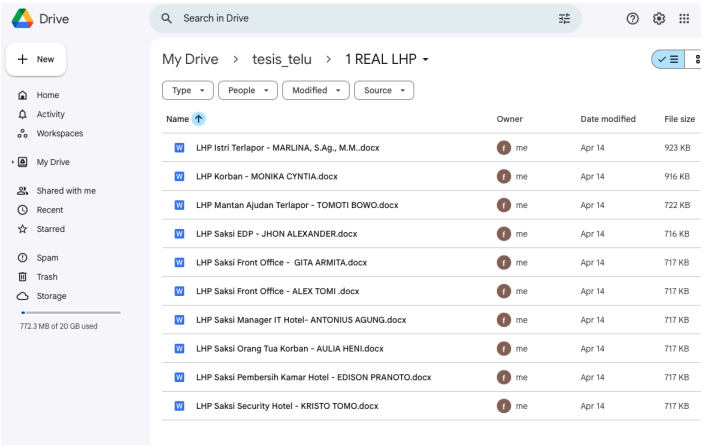


Figure 3. Police Report Documents on Google Drive

Documents that were originally in physical form must be converted to a digital format that can be processed by machines. This program supports the .docx format. Upload all Investigation Report documents or Investigations whose information will be extracted later to Google Drive (see Figure 6). Ensure that the AnatomiCrime Extractor.ipynb file and the Police Report file are in the same Google Drive account.

C. Connect Google Colab and Google Drive

1. Step 1 : import Drive module

Taking the drive object from the google.colab module, this object provides the functionality to mount Google Drive to the Colab file system so that it can be accessed via the path

2. Step 2 : installing Google Drive

Instructs Colab to mount the entire contents of Google Drive to the /content/drive folder on the Colab virtual machine. When run, Colab will display a Google authorization link; once you log in and grant permission, Drive will be mounted. Once successful, files and folders in Google Drive will be accessible like local directories, for example: '/content/drive/MyDrive/...' for reading/writing datasets, models, or documents.

```
from google.colab import drive
drive.mount('/content/drive')
```

3. folder_path is the name of a Python variable that will store the folder address. Its value "drive/MyDrive/tesis_telu/1 REAL LHP" is the relative path to the folder in your Google Drive containing the LHP (investigation report) files to be processed.

```
folder_path = "drive/MyDrive/tesis_telu/1 REAL LHP"
```

4. import glob, os: imports Python modules for file management and filename pattern searching. glob.glob() is used to create a list of all file paths whose names match a given pattern. f"{folder_path}/*.docx" is a path formatted to contain all files with the .docx extension from the target folder (e.g., drive/MyDrive/tesis_telu/1 REAL LHP).

```
import glob, os
file_paths = glob.glob(f"{folder_path}/*.docx")
```

5. os.listdir('/content/drive/MyDrive/tesis_telu/1 REAL LHP') Used to view the contents of a folder in Google Drive that has been mounted to Colab. The path '/content/drive/MyDrive/tesis_telu/1 REAL LHP' points to the folder in your Google Drive where the LHP document is stored.

```
os.listdir('/content/drive/MyDrive/tesis_telu/1 REAL LHP')
```

```
['LHP Istri Terlapor - MARLINA, S.Ag., M.M..docx',
 'LHP Korban - MONIKA CYNTIA.docx',
 'LHP Mantan Ajudan Terlapor - TOMOTI BOWO.docx',
 'LHP Saksi EDP - JHON ALEXANDER.docx',
 'LHP Saksi Front Office -  GITA ARMITA.docx',
 'LHP Saksi Front Office - ALEX TOMI .docx',
 'LHP Saksi Manager IT Hotel- ANTONIUS AGUNG.docx',
 'LHP Saksi Orang Tua Korban - AULIA HENI.docx',
 'LHP Saksi Pembersih Kamar Hotel - EDISON PRANOTO.docx',
 'LHP Saksi Security Hotel - KRISTO TOMO.docx']
```

Figure 9. Police Report File Listdir Results on Google Drive

D. WikiANN Dataset Installation

1. Step 1: Import the pandas library and give it the alias pd. Pandas is used to manipulate tabular data (DataFrames), for example, converting entity extraction results into tables, saving them to CSV, or performing simple statistical analysis.
2. Step 2: Import the load_dataset function from the datasets library (Hugging Face Datasets). load_dataset is used to conveniently load datasets, both public datasets on Hugging Face Hub (e.g., WikiANN) and local files such as CSV/JSON, so they can be directly used for training or evaluating NLP models.

```
import pandas as pd
from datasets import load_dataset
```

3. load_dataset(...) is a function from the datasets library (Hugging Face) that downloads and loads a dataset into memory. The first argument, "wikiann," specifies the name of the dataset to be retrieved, namely WikiANN, a multilingual corpus for Named Entity Recognition. The second argument, "id," is the language code for Indonesian, so that only the Indonesian-language subset of WikiANN is retrieved, not other languages.

```
dataset = load_dataset("wikiann", "id")
print(dataset)
```

These lines of code convert the three Hugging Face dataset splits (train, validation, test) into three easy-to-process pandas tables (DataFrames). The three pandas tables are:

a. train_df = pd.DataFrame(dataset['train']): Takes the train split from the dataset object (WikiANN in Indonesian) and wraps it into a DataFrame named train_df
b. valid_df = pd.DataFrame(dataset['validation']): Does the same for the validation split, so you have a valid_df containing the validation data in tabular format.
c. test_df = pd.DataFrame(dataset['test']): Converts the test split into a test_df, which stores the test data for the final model evaluation.

```
train_df = pd.DataFrame(dataset['train'])
valid_df = pd.DataFrame(dataset['validation'])
test_df = pd.DataFrame(dataset['test'])
```

4. The code below is used to check how much data is in each split of the dataset and displays it as a summary. The variables train_size, valid_size, and test_size store the number of samples in the training, validation, and testing subsets by calculating the length (len(...)) of each split.
   After that, three lines of print(...) write to the screen the amount of training, validation, and testing data so that the user can know the composition of the dataset before training and evaluating the model.

```
train_size = len(dataset['train'])
valid_size = len(dataset['validation'])
test_size = len(dataset['test'])
```

```
print(f"\nJumlah data training: {train_size}")
print(f'Jumlah data validation: {valid_size}")
print(f'Jumlah data testing: {test_size}")
```

```
Jumlah data training: 20000
Jumlah data validation: 10000
Jumlah data testing: 10000
```

Figure 4. WikiANN Dataset Composition

E. Convert .docx to .csv Documents

The code below automatically reads all .docx files in a folder, extracts the contents of each document by paragraph, and then combines the text of each document into a single line. The extracted results, containing the file names and the entire document contents, are saved as a DataFrame and then exported to a CSV file for easy analysis or further processing. This way, you can collect the contents of many legal documents in .csv format.

```
from docx import Document

def read_docx_content(file_path):
    document = Document(file_path)
    full_text = [para.text for para in document.paragraphs if para.text.strip() != ""]
    return full_text

folder_path = "drive/MyDrive/tesis_telu/1 REAL LHP"

file_paths = glob.glob(f"{folder_path}/*.docx")

full_text_data = []
paragraph_data = []

for file_path in file_paths:
    content = read_docx_content(file_path)

    full_text_data.append({
        "File Name": file_path,
        "Full Text": " ".join(content)
    })

    for para in content:
        paragraph_data.append({
            "File Name": file_path,
        })

df_full_text = pd.DataFrame(full_text_data)

full_text_file = "Hasil Konversi .docx to .txt.csv"

df_full_text.to_csv(full_text_file, index=False)

print(f"Hasil Konversi telah disimpan ke file: {full_text_file}")
```

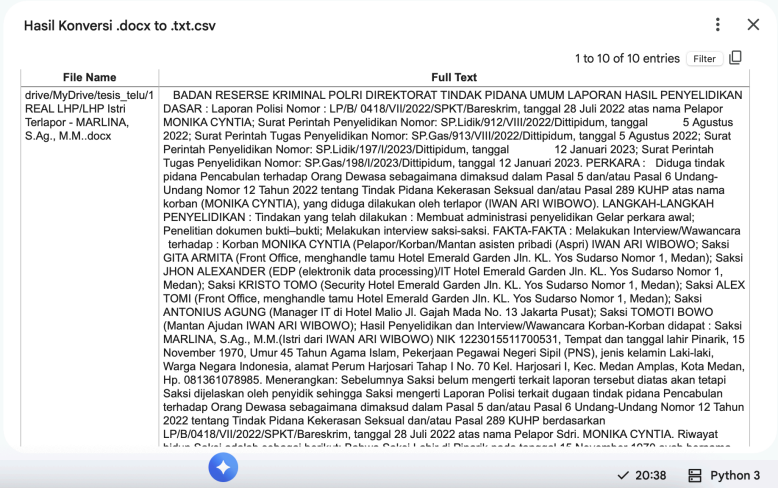


Figure 5. docx to .csv Conversion Results

F. Creating an Indonesian language NER model based on XLM-RoBERTa from Hugging Face

The code below is for creating an Indonesian NER based on Hugging Face's XLM-RoBERTa.

tokenizer = AutoTokenizer.from_pretrained("cahya/xlm-roberta-large-indonesian-NER") contains a tokenizer that has been fine-tuned to Cahyadi Hugging Face's Indonesian NER model, so that Indonesian text can be broken down into tokens in a manner consistent with the model.

model = AutoModelForTokenClassification.from_pretrained("cahya/xlm-roberta-large-indonesian-NER") contains the NER model weights that have been fine-tuned for the task of token classification (tagging entities) in Indonesian text, making it ready to be used to detect entities such as people, organizations, locations, and so on.

```
from transformers import AutoTokenizer, AutoModelForTokenClassification

##token -> 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

tokenizer = AutoTokenizer.from_pretrained("cahya/xlm-roberta-large-indonesian-NER")

model = AutoModelForTokenClassification.from_pretrained("cahya/xlm-roberta-large-indonesian-NER")
```

G. NER Pipeline in Indonesian

NER Pipeline Creates a handy function called nlp_ner for Named Entity Recognition (NER) tasks, which automatically combines the tokenizer and the model you previously loaded; when called with input text, nlp_ner will immediately tokenize, run the model, group subtokens into a single entity (because aggregation_strategy="simple"), and return a complete list of entities with their labels and scores, eliminating the need to manually write the technical steps of tokenization and inference.

```
from transformers import pipeline
```

```
nlp_ner = pipeline("ner", model=model, tokenizer=tokenizer, aggregation_strategy="simple")
```

H. Named Entity Recognition (NER) Process

The next step is an automated Named Entity Recognition (NER) process on all .docx documents in a folder, then summarizes the results into a CSV. In brief, the program reads the contents of each Word file, combines paragraphs into a single text, and then breaks the text into chunks of up to 512 tokens to fit the model's input limits.

Each chunk is returned to text, analyzed with the NER pipeline, and all detected entities (words, entity labels, label meanings, and scores) are stored in a list. Once all files have been processed, the NER results are converted to a DataFrame, the frequency of each entity type is calculated (using a Counter), and a summary table is created containing "Entity," "Entity Meaning," and "Count." Finally, the program saves the detailed NER results and the number of each entity to two CSV files, and displays a summary of the entity counts on screen. The following is a collection of NER labels that will be available in this program:

```
label_meanings = {
    "PER": "Nama orang (Person)",
    "ORG": "Organisasi (Organization)",
    "LOC": "Lokasi (Location)",
    "DAT": "Tanggal (Date)",
    "NOR": "Kewarganegaraan atau Kelompok Agama (Nationality or Religious Group)",
    "MISC": "Informasi lainnya (Miscellaneous)",
    "ORD": "Ordinal (Ordinal)",
    "GPE": "Entitas Geopolitik (Geopolitical Entity)",
    "MON": "Nilai Moneter (Monetary)",
    "PRD": "Produk (Product)",
    "QTY": "Kuantitas (Quantity)",
    "CRD": "Angka Kardinal (Cardinal Number)",
    "LAW": "Hukum (Law)",
    "REG": "Wilayah atau Peraturan (Region/Regulation)",
    "TIM": "Waktu (Time)",
    "FAC": "Fasilitas (Facility)",
    "LAN": "Bahasa (Language)",
    "EVT": "Peristiwa (Event)",
    "O": "Tidak ada entitas yang dikenali"
}
```


ner_hasil.csv

1 to 10 of 3149 entries Filter

File	Word	Entity	Entity Meaning	Score
drive/MyDrive/tesis_telu/1 REAL LHP/LHP Istri Terlapor - MARLINA, S.Ag., M.M..docx	BA	NOR	Kewarganegaraan atau Kelompok Agama (Nationality or Religious Group)	0.9946999
drive/MyDrive/tesis_telu/1 REAL LHP/LHP Istri Terlapor - MARLINA, S.Ag., M.M..docx	DAN RESERSE KRIMINAL POLRI DIREKTORAT TINDAK PIDANA UM	NOR	Kewarganegaraan atau Kelompok Agama (Nationality or Religious Group)	0.9886296
drive/MyDrive/tesis_telu/1 REAL LHP/LHP Istri Terlapor - MARLINA, S.Ag., M.M..docx	Laporan Polisi Nomor : LP/B/ 0418/VII/2022/SPKT/Bareskrim	LAW	Hukum (Law)	0.92120695
drive/MyDrive/tesis_telu/1 REAL LHP/LHP Istri Terlapor - MARLINA, S.Ag., M.M..docx	28 Juli 2022	DAT	Tanggal (Date)	0.9848118
drive/MyDrive/tesis_telu/1 REAL LHP/LHP Istri Terlapor - MARLINA, S.Ag., M.M..docx	MONIKA CYNTIA	PER	Nama orang (Person)	0.9565256
drive/MyDrive/tesis_telu/1 REAL LHP/LHP Istri Terlapor - MARLINA, S.Ag., M.M..docx	Surat Perintah Penyelidikan Nomor: SP.Lidik/912/VIII/2022/Dittipidum	LAW	Hukum (Law)	0.98735267
drive/MyDrive/tesis_telu/1 REAL LHP/LHP Istri Terlapor - MARLINA, S.Ag., M.M..docx	5 Agustus 2022	DAT	Tanggal (Date)	0.960812
drive/MyDrive/tesis_telu/1 REAL LHP/LHP Istri Terlapor - MARLINA, S.Ag., M.M..docx	Surat Perintah Tugas Penyelidikan Nomor: SP.Gas/913/VIII/2022/Dittipidum	LAW	Hukum (Law)	0.9899383
drive/MyDrive/tesis_telu/1 REAL LHP/LHP Istri Terlapor - MARLINA, S.Ag., M.M..docx	5 Agustus 2022	DAT	Tanggal (Date)	0.95282054
drive/MyDrive/tesis_telu/1 REAL LHP/LHP Istri Terlapor - MARLINA, S.Ag., M.M..docx	Surat Perinta	LAW	Hukum (Law)	0.9897462

✓ 20:38 Python 3

Figure 6. Entity Results from Each Document and Their Accuracy Score

ner_jumlah_entitas.csv

1 to 17 of 17 entries Filter

Entity	Entity Meaning	Count
NOR	Kewarganegaraan atau Kelompok Agama (Nationality or Religious Group)	34
LAW	Hukum (Law)	224
DAT	Tanggal (Date)	243
PER	Nama orang (Person)	1221
ORG	Organisasi (Organization)	285
LOC	Lokasi (Location)	291
GPE	Entitas Geopolitik (Geopolitical Entity)	261
CRD	Angka Kardinal (Cardinal Number)	150
QTY	Kuantitas (Quantity)	32
REG	Wilayah atau Peraturan (Region/Regulation)	20
ORD	Ordinal (Ordinal)	20
FAC	Fasilitas (Facility)	5
PRD	Produk (Product)	294
TIM	Waktu (Time)	30
MON	Nilai Moneter (Monetary)	31
LAN	Bahasa (Language)	2
EVT	Peristiwa (Event)	6

Show 100 per page

Figure 7. Entity Collection and Total Number of Findings

I. Regular Expression (Regex)

A regular expression (regex) is a text pattern used to search, extract, or modify substrings that meet a specific pattern within a string. In Python, regex is implemented through the re module with functions such as re.search, re.match, re.findall, re.sub, and re.split. In short, regex acts as a rule-based pattern matcher to transform long report text into structured pieces of information.

a. Pelapor

```
p = rx1(r"(?:Pelapor atas nama|atas nama Pelapor|yang melaporkan(?:[^:]*):?)\s*([A-Z][A-Z\s]+)", text)
```

b. Korban

```
k = rx1(r"(?:korban adalah|atas nama korban|korban(?:[^:]*):?)\s*\(?([A-Z][A-Z\s]+)\)?", text)
```

c. Terlapor

```
t = rx1(r"(?:terlapor atas nama|terlapor \(|yang diduga dilakukan oleh terlapor)\s*\(?([A-Z][A-Z\s]+)\)?", text)
```

d. Tempus Delicti (Tanggal Kejadian)

```
DATE_REGEXES = [

r"\b(\d{1,2}\s+(?:januari|februari|maret|april|mei|juni|juli|agustus|september|oktober|november|desember)\s+\d{2,4})\b",

r"\b(\d{1,2}\s+(?:january|february|march|april|may|june|july|august|september|october|november|december)\s+\d{2,4})\b",
    r"\b(\d{1,2}[/-]\d{1,2}[/-]\d{2,4})\b",
    r"\b(?:pada\s+tanggal|tanggal)\s+(\d{1,2}\s+\w+\s+\d{2,4})\b"
]

POS_KEYS = ["pada tanggal","tanggal kejadian","kejadian","terjadi","peristiwa","tempus delicti","tempus delicty","waktu kejadian"]

NEG_KEYS = ["tanggal lahir","lahir","laporan","lp","berita acara","bap","dicetak","ditandatangani","register","agenda","spdp","sprindik","pemeriksaan","surat"]

def score_date_occ(occ):
    ctx_low = occ["ctx"].lower()
    score = 0
    for k in POS_KEYS:
```

```
        if k in ctx_low:
            score += 4 if k in ("tempus delicti","tempus delicty","tanggal kejadian","waktu kejadian") else 3
    for k in NEG_KEYS:
        if k in ctx_low:
            score -= 5 if k in ("tanggal lahir","lahir","berita acara","bap","lp") else 2
    if re.search(r"\b20\d{2}\b", occ["norm"]):
        score += 1
    return score
```

e. Objek

```
VIOLENCE_KEYWORDS = [
"kekerasan","penganiayaan","aniaya","pemukulan","memukul","menendang","kdrt",
"perkosaan","pemerkosaan","pencabulan","pengancaman","penyekapan","pencekikan",
"penghinaan","perundungan","bullying","pemerasan","penyerangan"
```

f. Prasangka Pasal

```
pas = rx1(r"(Pasal\s+\d+[^;\n\.]*(?:KUHP|Undang-Undang Nomor[^;\n\.]*))", text)
```

g. Locus delicti (Alamat / TKP)

```
LOC_KEYS = ["lokasi","tempat","kejadian","tkp","alamat","jalan","jln","hotel","jl."]
```

h. Modus Operandi

```
MODUS_KEYS = ["modus","dengan cara","cara","maksud","niat","tujuan"]
```

i. Barang Bukti

```
BB_ITEM_PAT = re.compile(
    r"\b("
    r"video(?: [a-z0-9]+){0,2}|foto(?: [a-z0-9]+){0,2}|rekaman suara|voice note|chat
whatsapp|whatsapp|dokumen(?: [a-z0-9]+){0,2}|"

    r"kuitansi|kwitansi|nota|surat(?: [a-z0-9]+){0,2}|buku tabungan|rekening|kartu
atm|atm|hp|handphone|telepon|"

    r"uang(?: [a-z0-9\.]+){0,2}|minuman(?: [a-z0-9]+){0,2}|obat(?: [a-z0-9]+){0,2}|senjata(?: [a-z0-9]+){0,2}|"

    r"pisau|pakaian|baju|kondom|flash
?disk|flashdisk|laptop|komputer|cpu|harddisk|ssd|kamera|mic|microphone|speaker"

    r")\b", re.IGNORECASE
)

BB_SECTION_KEYS = ["barang bukti","bukti yang disita","disita","dokumen","whatsapp","video","voice
note","rekaman","foto","gambar"]
```

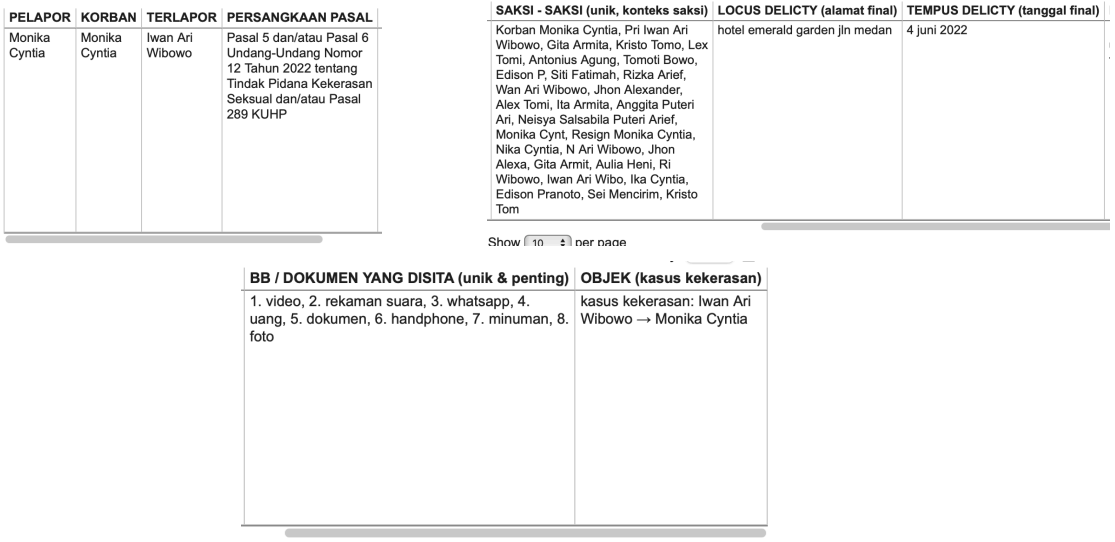
j. Saksi-Saksi

```
NAME_PREFIX = r"^(sdr|sdri|sdra|bapak|bpk|ibu|ib|ny|ny\.|nona|tuan|saudara|saksi|an|an\.)\b\.?:?\s*"
TRAIL_TRIM = [
    r"\bLahir.*",
    r"\btanggal\s+\d{1,2}\s+\w+\s+\d{2,4}",
    r"\d{1,2}[-/]\d{1,2}[-/]\d{2,4}",
    r"\btempat/tgl\b.*",
    r"\balamat\b.*",
]
```

```
WIT_REGEX = [ r"\bSaksi\s*(?:\d+|[IVXLCDM]+)?\s*[:\-]\s*([A-Z][A-Za-zÀ-ÖØ-öø-ÿ' .\-]+)",
r"\bSAKSI\s*(?:\d+|[IVXLCDM]+)?\s*[:\-]\s*([A-Z][A-Za-zÀ-ÖØ-öø-ÿ' .\-]+)",
r"Melakukan\s+Interview\/Wawancara\s+terhadap\s*[:\-]?\s*([A-Z][A-Za-zÀ-ÖØ-öø-ÿ' .\-]+)" ]
```

J. Program Output (Anatomy of Cryme)

The output of the Anatomy of Crime program is an automatic summary in the form of one row of data that condenses the core of a criminal case from a collection of Police Reports into structured columns, which include the identity of the reporter, victim, reported, article of suspicion, brief modus operandi, list of witnesses, locus delicti (incident address), tempus delicti (incident date), evidence or important documents confiscated, and the object of the case, so that law enforcers or analysts can see the "anatomy" of a crime quickly, consistently, and ready to be used as material for the initial case title.



PELAPOR	KORBAN	TERLAPOR	PERSANGKAAN PASAL
Monika Cyntia	Monika Cyntia	Iwan Ari Wibowo	Pasal 5 dan/atau Pasal 6 Undang-Undang Nomor 12 Tahun 2022 tentang Tindak Pidana Kekerasan Seksual dan/atau Pasal 289 KUHP

SAKSI - SAKSI (unik, konteks saksi)	LOCUS DELICTY (alamat final)	TEMPUS DELICTY (tanggal final)
Korban Monika Cyntia, Pri Iwan Ari Wibowo, Gita Armita, Kristo Tomo, Lex Tomi, Antonius Agung, Tomoti Bowo, Edison P, Siti Fatimah, Rizka Arief, Wan Ari Wibowo, Jhon Alexander, Alex Tomi, Ita Armita, Anggita Puteri Ari, Neisya Salsabila Puteri Arief, Monika Cynt, Resign Monika Cyntia, Nika Cyntia, N Ari Wibowo, Jhon Alexa, Gita Armit, Aulia Heni, Ri Wibowo, Iwan Ari Wibo, Ika Cyntia, Edison Pranoto, Sei Mencirim, Kristo Tom	hotel emerald garden jln medan	4 juni 2022

Show 10 per page

BB / DOKUMEN YANG DISITA (unik & penting)	OBJEK (kasus kekerasan)
1. video, 2. rekaman suara, 3. whatsapp, 4. uang, 5. dokumen, 6. handphone, 7. minuman, 8. foto	kasus kekerasan: Iwan Ari Wibowo → Monika Cyntia

Figure 8. Anatomy of Crime

K. Presentation of Data

The image presents a comparative evaluation of three machine learning models—Support Vector Machine (SVM), Random Forest, and Naive Bayes—applied for Named Entity Recognition (NER) on legal documents. The evaluation is based on metrics such as precision, recall, F1-score, and confusion matrices for each model.

1. NER Without Regex

Based on the experimental evaluation, the performance of three classification algorithms—Support Vector Machine (SVM), Naive Bayes, and Random Forest—was assessed using Precision, Recall, and F1-Score as evaluation metrics, as summarized in the experimental results table.

The Support Vector Machine (SVM) achieved a precision of 0.8095, a recall of 0.85, and an F1-score of 0.8293. These results indicate that SVM provides a balanced classification performance, maintaining a relatively high level of accuracy in predicting positive instances while also achieving good sensitivity toward relevant data. The balance between precision and recall is reflected in its F1-score, suggesting that SVM is a stable and reliable model for the classification task in this study.

The Naive Bayes classifier obtained a precision of 0.8125, a recall of 0.65, and an F1-score of 0.7222. Although its precision is slightly higher than that of SVM, the noticeably lower recall indicates that the model fails to identify a substantial portion of positive instances. This limitation is likely attributable to the strong conditional independence assumption inherent in the Naive Bayes model, which may reduce its effectiveness when dealing with complex or correlated features. Consequently, the lower recall adversely impacts the overall F1-score, making Naive Bayes the weakest-performing model among the three.

In contrast, the Random Forest algorithm demonstrates the strongest overall performance, achieving a precision of 0.80, a perfect recall of 1.00, and the highest F1-score of 0.8889. The recall value of 1.00 indicates that the model successfully identifies all positive instances, resulting in zero false negatives. Although its precision is marginally lower than those of SVM and Naive Bayes, the combination of very high recall and strong precision leads to the highest F1-score. This result suggests that Random Forest is highly effective at capturing complex patterns and feature interactions within the data.

Overall, considering the F1-score as a balanced measure of classification performance, Random Forest outperforms SVM and Naive Bayes in this study. These findings suggest that ensemble-based approaches, such as Random Forest, are better suited for handling data complexity and variability compared to probabilistic classifiers and margin-based linear models.

Table 1. NER evaluation matrix without regex

Algorithm	Precision	Recall	F1-Score
SVM	0.809524	0.85	0.829268
Naive Bayes	0.812500	0.65	0.722222
Random Forest	0.800000	1.00	0.888889

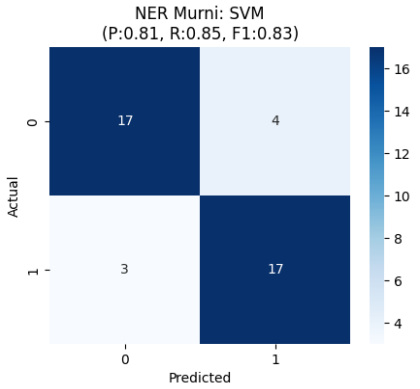


Figure 9. Confusion Matrix SMV

Figure 9 presents the confusion matrix of the pure NER model using Support Vector Machine (SVM). The model correctly classified 17 negative instances (true negatives) and 17 positive instances (true positives). However, there were 4 false positives, where non-entity tokens were incorrectly classified as entities, and 3 false negatives, where actual entity tokens were not successfully identified. These misclassifications contribute to a precision of 0.81 and a recall of 0.85, resulting in an overall F1-score of 0.83. The results indicate that while

the SVM-based NER model demonstrates balanced performance, it still encounters challenges in distinguishing entity boundaries, particularly in ambiguous token contexts.

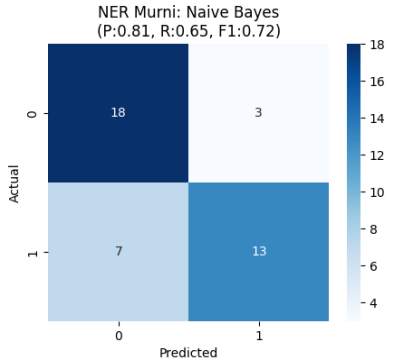


Figure 10. Confusion Matrix Naïve Bayes

Figure 10 illustrates the confusion matrix of the pure NER model using Naive Bayes. The model correctly identified 18 negative instances (true negatives) and 13 positive instances (true positives). However, it produced 3 false positivesand a relatively high number of 7 false negatives, indicating that several entity tokens were not successfully detected. This behavior results in a precision of 0.81 and a lower recall of 0.65, leading to an overall F1-score of 0.72. The results suggest that while Naive Bayes maintains reasonable precision, it struggles to capture entity occurrences consistently, primarily due to its strong independence assumptions between features.

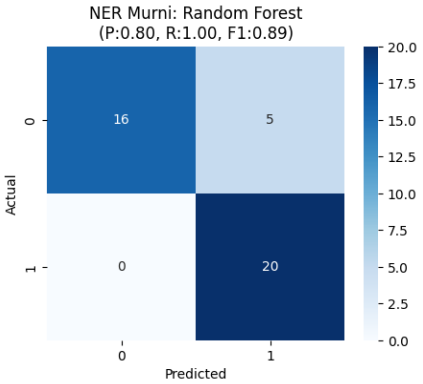


Figure 11. Confusion Matrix Random Forest

Figure 11 shows the confusion matrix of the pure NER model using Random Forest. The model correctly classified 16 negative instances (true negatives) and 20 positive instances (true positives), with no false negatives, indicating perfect recall. However, 5 false positives were observed, where non-entity tokens were incorrectly labeled as entities. This results in a precision of 0.80, a recall of 1.00, and an F1-score of 0.89, demonstrating that the Random Forest model is highly effective in identifying all entity tokens, albeit with a moderate tendency toward over-identification.

2. NER + Regex (Hybrid)

The performance of the three classification algorithms—Support Vector Machine (SVM), Naive Bayes, and Random Forest—was evaluated using Precision, Recall, and F1-Score as quantitative performance metrics, as presented in the experimental results.

The Support Vector Machine (SVM) achieved perfect scores across all evaluation metrics, with a precision, recall, and F1-score of 1.000. This result indicates that the model correctly classified all instances without producing any false positives or false negatives within the evaluation dataset. Such performance suggests that the feature space is highly separable under the SVM decision boundary and that the learned model aligns exceptionally well with the underlying data distribution.

Similarly, the Random Forest classifier also attained perfect performance, achieving 1.000 for precision, recall, and F1-score. This indicates that the ensemble-based approach was able to fully capture the discriminative patterns in the data, effectively handling feature interactions and decision boundaries without classification errors. The identical performance between SVM and Random Forest suggests that both linear margin-based and ensemble tree-based methods are equally effective for this specific dataset and task configuration.

In contrast, the Naive Bayes classifier achieved a precision of 0.8947, a recall of 0.9189, and an F1-score of 0.9067. While its performance remains strong, the slightly lower scores compared to SVM and Random Forest indicate the presence of both false positives and false negatives. This performance gap can be attributed to the conditional independence assumption inherent in Naive Bayes, which may limit its ability to model complex feature dependencies present in the dataset.

Although the perfect performance achieved by SVM and Random Forest indicates excellent classification capability, such results should be interpreted with caution. Perfect evaluation scores may suggest that the dataset is relatively small, highly structured, or exhibits limited variability, which can potentially lead to overfitting. Therefore, further validation using larger datasets, cross-domain samples, or cross-validation techniques is necessary to confirm the robustness and generalizability of the proposed models.

Overall, based on the F1-score as a balanced metric, SVM and Random Forest demonstrate superior performance compared to Naive Bayes in this study. These findings indicate that discriminative and ensemble-based models are more suitable for the classification task under the current experimental setting.

Table 2. NER evaluation matrix with regex

Algorithm	Precision	Recall	F1-Score
SVM	1.000000	1.000000	1.000000
Naive Bayes	0.894737	0.918919	0.906667
Random Forest	1.000000	1.000000	1.000000

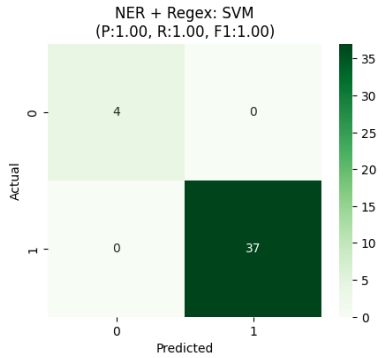


Figure 12. Confusion Matrix NER + Regex SVM

Figure 12 illustrates the confusion matrix of the NER + Regex model using Support Vector Machine (SVM). The model correctly classified all negative instances (4 true negatives) and all positive instances (37 true positives), with no false positives and no false negatives. This results in perfect performance with precision, recall, and F1-score all equal to 1.00. The findings indicate that the integration of rule-based regular expressions as a post-processing layer effectively resolves residual misclassifications from the pure NER model, leading to complete and consistent entity recognition in the evaluated dataset.

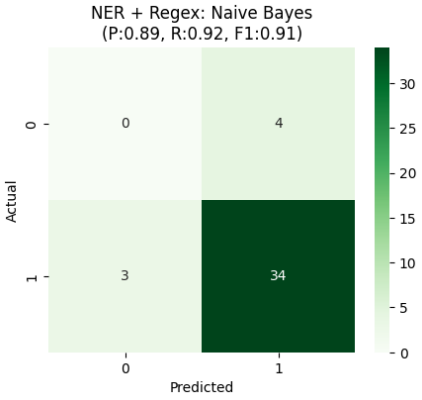


Figure 13. Confusion Matrix NER + Regex Naive Bayes

Figure 13 presents the confusion matrix of the NER + Regex model using Naive Bayes. The model correctly identified 34 positive instances (true positives) but failed to correctly classify negative instances, resulting in 4 false positives and 3 false negatives, with no true negatives observed. This performance yields a precision of 0.89, a recall of 0.92, and an F1-score of 0.91. The results indicate that the incorporation of regular expression rules significantly improves the recall of the Naive Bayes-based NER model, although some over-identification of entities remains.

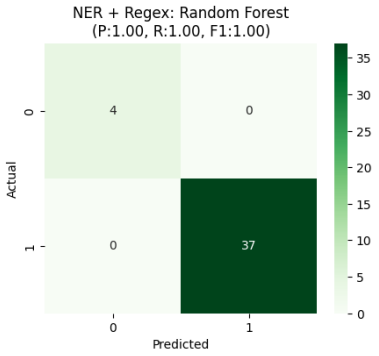


Figure 14. Confusion Matrix NER + Regex Random Forest

Figure 14 shows the confusion matrix of the NER + Regex model using Random Forest. The model correctly classified all negative instances (4 true negatives) and all positive instances (37 true positives), with no false positives and no false negatives. Consequently, the model achieves perfect performance with precision, recall, and F1-score all equal to 1.00. This result demonstrates that the combination of an ensemble-based NER model with rule-based regular expression post-processing effectively eliminates residual classification errors, leading to highly accurate and consistent entity recognition.

L. Summary of Findings

This study addressed the increasingly complex challenge of managing and analyzing large volumes of legal documents within the Indonesian National Police (Polri). The research aimed to develop and evaluate an automated system integrating Named Entity Recognition (NER) and rule-based Regular Expressions (Regex) within a BERT-based transformer framework to support the creation of preliminary investigation reports.

The research employed an experimental design involving fine-tuning of pre-trained transformer models and the application of NER pipelines evaluated with and without Regex-based post-processing. The primary objective was to assess the effectiveness of NER models in identifying key legal entities—such as suspects, victims, time, location, and legal article references—and to examine how Regex rules contribute to improving entity consistency and completeness.

The experimental results demonstrate a clear performance improvement when Regex is incorporated into the NER pipeline. As shown in Table 27, NER models without Regex achieved moderate to high performance, with F1-scores of 0.83 (SVM), 0.72 (Naive Bayes), and 0.89 (Random Forest). While these results indicate reasonable baseline capability, errors such as false positives and false negatives remained evident, particularly in handling structured legal expressions and format variations.

In contrast, the results presented in Table 28 show that the integration of Regex significantly enhances NER performance. Both SVM and Random Forest achieved perfect scores (Precision = Recall = F1 = 1.00), while Naive Bayes also exhibited substantial improvement, reaching an F1-score of 0.91. These findings confirm that Regex effectively resolves residual ambiguities left by statistical and machine-learning-based NER models, particularly in recognizing standardized legal patterns such as article numbers, dates, and formal entity references.

The superiority of the NER + Regex approach indicates that combining data-driven learning with domain-specific linguistic rules provides a more robust and reliable solution for legal text processing. Regex functions as a corrective post-processing layer that enforces structural constraints, thereby reducing misclassification errors and improving entity boundary precision.

Overall, the findings suggest that NER enhanced with Regex consistently outperforms pure NER models, making it more suitable for operational deployment in law enforcement contexts. This hybrid approach not only improves accuracy but also increases the reliability and interpretability of extracted entities, thereby supporting faster, more consistent, and more standardized investigation report formulation within Polri.

CONCLUSION

This study concludes that the automated investigation report generation system integrating BERT-based Named Entity Recognition (NER) and Regular Expressions (Regex) is effective in processing Indonesian legal documents, as evidenced by high precision, recall, and F1-scores achieved by IndoBERT, XLM-R, and RoBERTa; the combined NER and Regex approach strengthens entity extraction accuracy, improves structural consistency, and outperforms purely model-based NER, enabling reliable identification of key legal information such as suspects, victims, time, location, evidence, and legal articles while supporting concise and context-preserving document summarization. Overall, these findings indicate that NLP-based systems can significantly reduce manual workload, minimize bias, enhance analytical reliability, and support digital transformation within Polri; therefore, future research should expand datasets to include broader legal document varieties, collaborate with legal experts for annotation refinement, optimize computational efficiency, integrate advanced legal analytics such as retrieval or clustering, and encourage institutional adoption. Additionally, future development can extend this prototype into a fully integrated NER-aware summarization framework where extracted entities guide summary generation, evaluated using both automatic metrics and expert legal assessments to further improve practicality and legal robustness. For future research, it is recommended to expand datasets to cover more diverse legal document types, involve legal experts to refine annotation quality, improve computational efficiency, and integrate advanced legal analytics such as retrieval and clustering methods.

REFERENCES

Akerele, J. I., Uzoka, A., Ojukwu, P. U., & Olamijuwon, O. J. (2024). Increasing software deployment speed in agile environments through automated configuration management. *International Journal of Engineering Research Updates*, *7*(02), 28–35.

Alevizos, L. (2025). Automated cybersecurity compliance and threat response using AI, blockchain and smart contracts. *International Journal of Information Technology*, *17*(2), 767–781.

Allioui, H., & Mourdi, Y. (2023). Exploring the full potentials of IoT for better financial growth and stability: A comprehensive survey. *Sensors*, *23*(19), 8015.

Alqahtani, F. M., Noman, M. A., Alabdulkarim, S. A., Alharkan, I., Alhaag, M. H., & Alessa, F. M. (2023). A new model for determining factors affecting human errors in manual assembly processes using fuzzy delphi and DEMATEL methods. *Symmetry*, *15*(11),

1967.
Balcıoğlu, Y. S., Çelik, A. A., & Altındağ, E. (2024). Integrating blockchain technology in supply chain management: a bibliometric analysis of theme extraction via text mining. *Sustainability*, *16*(22), 10032.
Bernardi, F. A., Alves, D., Crepaldi, N., Yamada, D. B., Lima, V. C., & Rijo, R. (2023). Data quality in health research: integrative literature review. *Journal of Medical Internet Research*, *25*, e41446.
Bjelobaba, S., Waddington, L., Perkins, M., Foltýnek, T., Bhattacharyya, S., & Weber-Wulff, D. (2025). Maintaining research integrity in the age of GenAI: an analysis of ethical challenges and recommendations to researchers. *International Journal for Educational Integrity*, *21*(1), 18.
Černý, J., Avramov, K., & Pendse, L. R. (2026). A multi-stage agentic AI system for extracting information from large digital archives: case study on the Czechoslovak year 1968 in CIA's FOIA collection. *The Electronic Library*, 1–26.
Feldhoff, K., Wiemer, H., Träger, P., Kühne, R., Zimmermann, M., & Ihlenfeldt, S. (2025). Automatic Information Extraction from Scientific Publications Based on the Use Case of Additive Manufacturing. *Applied Sciences*, *15*(17), 9331.
Halford, E. (2026). Organizational Trauma Within Policing: A Case Study of the United Kingdom. *The Journal of Applied Behavioral Science*, *62*(1), 130–164.
Klarin, A. (2024). How to conduct a bibliometric content analysis: Guidelines and contributions of content co‐occurrence or co‐word literature reviews. *International Journal of Consumer Studies*, *48*(2), e13031.
Mahadevkar, S. V, Patil, S., Kotecha, K., Soong, L. W., & Choudhury, T. (2024). Exploring AI-driven approaches for unstructured document analysis and future horizons. *Journal of Big Data*, *11*(1), 92.
Marzi, G., Balzano, M., Caputo, A., & Pellegrini, M. M. (2025). Guidelines for bibliometric‐systematic literature reviews: 10 steps to combine analysis, synthesis and theory development. *International Journal of Management Reviews*, *27*(1), 81–103.
Morais, C., Yung, K. L., Johnson, K., Moura, R., Beer, M., & Patelli, E. (2022). Identification of human errors and influencing factors: A machine learning approach. *Safety Science*, *146*, 105528.
Salunkhe, A., Pawar, V., Pise, P., Mule, S., Survase, A., Godase, V., & Zambre, S. (2025). A Review on Real-Time RFID-Based Smart Attendance Systems for Efficient Record Management. *Advance Research in Analog and Digital Communications*, *2*(2), 32–46.
Shanthi, P. (2025). Smart Document Analysis and Validation System for Semi-Structured Data. *2025 International Conference on Recent Innovation in Science Engineering and Technology (ICRISET)*, 1–8.
Surahman, E., & Wang, T. (2022). Academic dishonesty and trustworthy assessment in online learning: A systematic literature review. *Journal of Computer Assisted Learning*, *38*(6), 1535–1553.
Tan, Y. S., Zalzuli, A. D., Ang, J., Ho, H. F., & Tan, C. (2022). Understanding the workload of police investigators: a human factors approach. *Journal of Police and Criminal Psychology*, *37*(2), 447–456.
Taquette, S. R., & Borges da Matta Souza, L. M. (2022). Ethical dilemmas in qualitative research: A critical literature review. *International Journal of Qualitative Methods*, *21*, 16094069221078732.
Walter, T., Korhonen, L., Otterman, G., van Agthoven, G., & Jud, A. (2025). Challenges to reliable ICD-10 coding of child maltreatment: A qualitative interview study of healthcare professionals in German and Swedish hospitals. *Child Abuse & Neglect*, *164*, 107446.