Universiteit Leiden
The Netherlands

Bachelor Computer Science & Datascience and Artificial Intelligence

News mining for Homicide characteristics in Indonesia

for a Thesis Research Proposal

Leïla Aïkili Hagen s3455521

First supervisor and second supervisor:
Marco Spruit, Olga Bogolyubova

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

21/02/2025

Abstract

We are currently experiencing an explosion of textual data, with news articles often containing valuable information about homicides. Extracting this information can support policymakers and forensic analysts in identifying patterns and developing effective crime prevention strategies. However, the large volume, unstructured nature of the text, and limited labeled data pose significant challenges. This study compares rule-based methods (RegEx) and embedding-based models (IndoBERT) for classifying homicide-related articles and extracting key incident characteristics. We also evaluate Named Entity Recognition (NER) and Relation Extraction (RE) models for extracting structured crime-related attributes. Using over 10,000 manually labeled Indonesian news articles, we evaluated models primarily via weighted F1-score. The RegEx model achieved 0.4918, while embedding-based models significantly outperformed it: simple embedding-based classifier 0.7949, weighted embedding-based classifier 0.8533, and weighted classifier with hyperparameter tuning 0.8606. For RE, the weighted F1-score was 0.59. NER performance improved from 0.2645 without data augmentation to 0.4474 with augmentation, and further to 0.5195 when combined with hyperparameter tuning, highlighting the value of data enrichment and model optimization. Overall, embedding-based models provide a scalable solution for classifying homicide content and extracting structured information from Indonesian news. NER models, enhanced with augmentation and tuning, outperform RE in extracting detailed incident attributes, emphasizing the importance of large datasets and careful handling of class imbalance in low-resource NLP tasks.

Contents

1 Introduction

1.1 Background & Context

Globally, overall crime rates, including homicide and violent crime, are decreasing. According to the UNODC Global Study on Homicide (2019 and 2023), homicide rates have steadily declined since 1990. This downward trend continued in the latest data from 2021. The same pattern appears in Indonesia, where the number of intentional homicide victims per 100,000 people dropped from 0.5 in 2017 to 0.3 in 2021. (UNODC, 2019, 2023).

However, Indonesia struggles with underreporting crime and sensationalized media coverage, which raises doubts about data accuracy. Police reluctance and bureaucratic obstacles often hinder official reporting. Many victims don't report crimes due to fear of retaliation or lack of trust in law enforcement. Indonesia also faces challenges in gender- and race-sensitive reporting (Musyafak et al., 2025). Violence against women and minority groups is often underreported or misrepresented in the media. This creates a strong bias in how these issues are portrayed and understood by the public (Norouzi, 2022).

Additionally, Indonesian media often dramatizes crime stories to attract readers. They sometimes exaggerate facts or focus too much on violent details, which can distort public understanding. Crime reporting lacks standard protocols, causing inconsistent data across media and official reports. This makes research and policy-making more difficult. Sensationalism can pressure law enforcement to act, but it also spreads misinformation and increases public fear (Musyafak et al., 2025).

According to the European Journal of Development Studies, Indonesia's low homicide rates may reflect limited capacity to report crime. This contrasts with countries like Brunei or Singapore, which have similarly low rates but stronger crime-reporting systems. Some studies support this, showing a 7% increase in crime in 2021. This rise is linked to higher unemployment and poverty in certain regions of Indonesia (Septriani, 2024; Rajagukguk, 2023).

Simultaneously, there has been an explosion of unstructured textual data in news reports. For example, in 2018, Indonesia had approximately 43,300 online media portal, and the average annual output per outlet was approximately 3,650 articles. Thus, Indonesia could be producing over 158 million online news articles annually (Prasetyo and Ajitrisna, 2023; Ulfatriyani et al., 2020; Musyafak et al., 2025). These large-scale textual datasets contain critical information about crime incidents and provide valuable insights into crime patterns. The digitization of news has created a vast and continuously updating source of crime-related information; however, given the massive volume of text, manual analysis would be incredibly time-consuming. (Tanwar et al., 2015)

1.2 Problem Statement

The volume of textual data explains the demand for automating text mining with natural language processing (NLP). NLP provides a powerful tool for extracting valuable information from vast

amounts of unstructured text data. Its applications range from sentiment analysis and automated summarization to complex tasks such as Named Entity Recognition (NER) and Relationship Extraction (RE). Importantly, NLP techniques have shown great promise in addressing social issues, such as monitoring crime reporting, detecting misinformation, and analyzing crime reports for improved law enforcement and policy making. (Sarzaeim et al., 2023).

Among the key NLP methods relevant to this research are regular expressions (RegEx), embedding-based classification, NER, and RE. RegEx is a rule-based approach that uses pattern matching to identify specific keywords or phrases within text. Although straightforward, RegEx often struggles with language variability and context, leading to limited accuracy. (Subowo et al., 2025; Pillar et al., 2022) Embedding-based classification leverages vector representations of words or sentences to capture semantic meaning, enabling more flexible and robust text classification. NER focuses on identifying and categorizing entities such as persons, locations, and weapons mentioned in text, which is critical for structuring unstructured news data. RE builds upon NER by detecting and classifying relationships between entities, facilitating the creation of meaningful information graphs (Chen et al., 2020). These deep learning model perform best with supervised learning, as such they require a long training procedure that is supported by labeled data (Bifari et al., 2024).

The greatest challenge in creating these text analysis models is gathering sufficient labeled data. A fair amount of research regarding NER has been done with English, Chinese, Spanish, and German. Subsequently, these resourceful languages have well well-developed, openly available NER datasets specialized in the crime domain (Bose and Sarkar, 2024). However, Indonesian remains a low-resource language. Studies specifically targeting Indonesian crime news are scarce, and there are limited open-access NER domains.

Aside for limited annotated datasets tailored to crime in the local Indonesian context, another obstacle the models face is handling noisy text data and informal language usage. Moreover, existing models often lack the necessary domain adaptation to accurately recognize entities and relationships unique to Indonesian legal and cultural settings (Subowo et al., 2025). Additionally, it is important to acknowledge the incompleteness of textual data, since news mining is limited by bias in media coverage. Sensationalist reporting can exaggerate crime threats, while under-reporting certain crimes can lead to public complacency (Prasetyo and Ajitrisna, 2023).

Addressing these gaps is essential for developing reliable tools that can transform raw news reports into actionable insights. By creating a structured dataset on crime in Indonesia, we can help policymakers and law enforcement develop effective strategies and interventions to prevent violent crimes. Furthermore, studying homicide can improve forensic methodologies and enhance the accuracy of criminal investigations. (Etzioni et al., 2011)

1.3 Research Objectives

The research aim is to develop a robust methodology for automatically identifying and extracting homicide-related information from Indonesian news articles using NLP techniques. This involves comparing traditional rule-based methods with modern model-based methods to classify and extract

information. The research addresses the challenge of transforming unstructured, noisy news data into structured crime information that can support policymakers and law enforcement. Therefore, the main research question is: "How can NLP techniques be used to automatically identify and extract homicide characteristics from Indonesian news articles?"

To answer this, the methodology is divided into two key steps. The first step focuses on identifying homicide-related articles through binary classification. This raises the sub-question: How does the performance of traditional rule-based approaches, such as RegEx, compare to modern embedding-based classification methods for detecting homicide-related news? This comparison will help determine the most effective approach for dealing with noisy and imbalanced datasets.

The second step involves extracting detailed homicide characteristics from the identified articles. This leads to the sub-question: How do NER methods compare to RE techniques in accurately identifying crime-related entities and relationships in news text? Addressing this will provide insights into the best strategies for capturing complex information from unstructured text.

Together, these objectives aim to evaluate methods of automating crime data extraction, ultimately, creating a scalable method to conduct text mining in the domain of homicide and improving the availability and quality of homicide data for further criminal analysis and policy-making.

1.4 Brief Methodology

The first step of the research was dataset construction. News scraping was performed on Indonesian articles, collecting one year's worth of news published by Kompas in 2024. The dataset was manually labeled by classifying articles as homicide-related or non-homicide-related. This created the necessary data for training, validation, and testing of classification models.

Both RegEx and embedding-based models were developed. The RegEx model was fine-tuned by adjusting keyword matching, while the embedding-based model, built using IndoBERT, was fine-tuned by modifying its learning parameters. To handle class imbalance, sampling techniques and weighted loss functions were applied. Final model performance was compared using accuracy, precision, recall, and F1-score.

The next phase focused on information extraction. NER and RE methods both using IndoBERT as the base model. The model performance was assessed using F1-score, precision, recall, and confusion matrices. Limitations were analyzed and potential improvements proposed. Finally, a CSV file was generated to represent each article along with its extracted homicide-related information.

1.5 Contribution & Limitations

The research focuses exclusively on homicide-related news articles published in Indonesian media outlets. To date, this is the first structured attempt to create a homicide-specific classification and extraction pipeline. Existing pipelines for Indonesian news data cover a broader range of crimes,

such as theft, fraud, or drug offenses. These were excluded to maintain specificity.

The developed pipelines are scalable and applicable to any dataset of Indonesian news articles. However, the models are limited to the Bahasa Indonesia language, as they are built on IndoBERT—a language-specific model—and trained on Indonesian domain-specific data. Applying the model to another language would require selecting and configuring a language-specific BERT model and obtaining a corresponding labeled dataset.

Although the model is not directly generalizable to other domains or languages, the findings may still inform future text analysis research beyond the Indonesian language or crime analysis. The research compares traditional and modern NLP techniques under real-world conditions, showing that when annotated data is limited, traditional methods may outperform advanced model-based approaches. It also highlights challenges in processing noisy, imbalanced data in a low-resource language. The use of explainability tools improves model interpretability, a feature often overlooked in NLP applications for social sciences.

In addition to training the selected models, the research produces a comprehensive and publicly available dataset consisting of all newspaper articles published by Kompas in 2024. From this corpus, two labeled datasets are created. The first categorizes each article as either homicide-related or not. The second focuses on homicide-related articles and includes detailed annotations of extracted homicide characteristics, identifying relevant entities and their relationships in a structured format.

As a result, the research contributes a chronologically organized and accessible archive of Kompas articles (stored in JSON format), along with a high-quality, manually annotated dataset. This labeled data is valuable for further development and refinement of homicide-specific news mining systems, as model performance depends heavily on the availability of domain-specific training data.

Finally, the trained model was applied on a broader set of articles to generate a structured CSV file. This file presents each article alongside its extracted homicide characteristics in an interpretable format. The structured representation facilitates the generation of crime statistics, the identification of trends or hotspots, and may provide actionable insights for law enforcement or policymakers.

1.6 Structure of the Thesis

This thesis is structured as follows. Chapter 2 provides the theoretical background, introducing key concepts such as NLP in the context of crime news mining, classification methods using RegEx and embeddings, techniques for handling imbalanced datasets, and approaches for information extraction through NER and RE. It also reviews relevant related work. Chapter 3 outlines the methodology, detailing the steps used for article classification and information extraction. Chapter 4 presents the data preprocessing steps and the experimental setup, including evaluation metrics. Chapter 5 reports the results of both the classification and information extraction tasks. Chapter 6 offers a critical discussion of the findings, including their implications and limitations. Finally, Chapter 7 concludes the thesis and suggests directions for future research.

2 Theoretical Background

This section provides the theoretical foundation for the thesis by reviewing existing approaches to crime-related text analysis using NLP. It covers traditional methods like RegEx, advanced techniques such as embedding-based classification, NER, and RE, and discusses their relevance and limitations in the context of Indonesian news data.

2.1 NLP in Crime News Mining

NLP is a subfield of artificial intelligence and computational linguistics. It is the science of enabling computers to understand, interpret, and generate human language. NLP techniques are designed to retrieve and translate raw text into structured representations. This method can be leveraged to extract meaningful patterns and provide a scalable solution to process large volumes of unstructured text data (Sarzaeim et al., 2023)

This process of extracting structured insights from unstructured news text is called *news mining*. In the field of criminology, it can be applied to identify crime patterns, track geographic trends, and recognize entities such as victims or suspects. In the early 2000s, researchers created the first implementations of NLP techniques in social science and journalism. They employed term frequency analysis and keyword matching for automated text classification. Although simple, these rule-based approaches laid the foundation for using language technology to systematically analyze crime-related news. AlatristaSalas2020.

2.2 RegEx Classification

One of the earliest methods for classifying crime-related texts is keyword-based filtering using RegEx. This approach can be used to determine whether an article reports on a crime and identify the type of crime. RegEx works by defining specific patterns, such as exact words or phrases, and then searching for them in the text. One of the main advantages of RegEx is that is has a fast execution and low computational power, thus you can quickly determine whether an article is homicide-related or not (Dharviyanti and Wilantika, 2024). Additionally, the model's algorithm is simple and transparent, making it easy to understand how classifications are made. Moreover, it does not require training on labeled data, which allows for easy adaptation and fine-tuning, since the only parameters are the predefined rules. (Ahmad et al., 2018; Umair et al., 2020)

However, one of the models' limitations is that the rules are fixed and strict and need to be manually changed. This means that the model does not learn on its own based on new textual data. It may thus struggle with generalization and not perform well on unseen data (Umair et al., 2020; Arulanandam et al., 2014). Aside from this, the configuration of the rules requires expert knowledge and an understanding of the language to be able to choose appropriate keywords and phrases for the classifier. It often requires high-precision keywords to minimize false positives, as broad

terms can return irrelevant results. For example, the keyword “stab” may retrieve a wide range of violent incidents not necessarily classified as homicide. Thus, selecting keywords involves linguistic considerations to ensure semantic precision (Sedik and Romadhony, 2023). Otherwise, it would be necessary to create a more complex RegEx that includes if, and, or statements, such as: if ”stab” and ”died,” then it is a homicide case.

In Indonesian, complex word forms create extra challenges. Verbs often appear with different affixes, like *membunuh* (“to kill”), *pembunuh* (“killer”), or *dibunuh* (“killed”). This makes it challenging to design keywords that are both complete and accurate.

Although RegEx provides a simple and effective means to classify text, it lacks a real understanding of language. The model does not recognize nuance or implied meaning unless it is explicitly defined in the rules. For example, if a euphemism like ”the person passed away” is used, the model will not identify it as a homicide unless that exact phrase is included in the rule set. Similarly, RegEx cannot distinguish context. A sentence like ”the player killed the game” may be wrongly flagged as a homicide case, even though the word ”killed” is not used in a violent or criminal sense. This limitation highlights how RegEx depends heavily on simple word matching and lacks the contextual awareness needed to handle more complex or ambiguous language. (Umair et al., 2020; Arulanandam et al., 2014).

2.3 Embedding-Based Classification

A more advanced approach to classifying articles involves embedding-based classification. This method uses word or sentence embeddings to represent text in a continuous vector space. Instead of relying only on keyword matching, embedding models convert words or entire texts into dense numerical vectors. These vectors capture semantic meaning and contextual relationships. Embeddings help the model understand subtle nuances in language. They also allow the model to recognize patterns beyond exact word matches. For instance, words with similar meanings or related contexts are mapped to vectors that are close together in the embedding space. (Alatrista-Salas et al., 2020; Sarzaeim et al., 2023; Sulastri et al., 2023)

Classification algorithms, such as neural networks or support vector machines, then use these vectors as input. This allows the model to categorize articles according to crime type. This approach typically improves performance, especially in complex or noisy texts. It captures deeper linguistic features that rule-based methods may miss (Alatrista-Salas et al., 2020; Bifari et al., 2024; Chen et al., 2020; Ku and Leroy, 2014)

Since we are handling an Indonesian corpus, we utilized IndoBERT. Many successful implementations of crime analysis on Indonesian unstructured text utilize it due to its strong ability to capture contextual and semantic nuances specific to the Indonesian language (Pongpaichet et al., 2024; Silalahi et al., 2022; Subowo et al., 2025).

2.4 IndoBERT Architecture and Fine-Tuning Challenges

IndoBERT is a pre-trained Transformer-based monolingual language model tailored for Indonesian. The model was trained on a large Indonesian corpus including Wikipedia and news. IndoBERT generates contextualized embeddings that capture semantics and polysemy better than traditional methods. (Wilie et al., 2020) Unlike RNNs or LSTMs, Transformers use self-attention to capture relationships between words without relying on recurrence. IndoBERT follows BERT's architecture, which is consistent with most Transformer models and starts with tokenization. This process starts by breaking text into smaller units called tokens, which usually consist of words and subwords. This step is necessary since deep learning models operate on numerical input. Therefore, to prepare the textual data for neural models, padding and truncation have to be performed. Padding is a procedure that ensures all sequences have the same length by adding special characters at the end of a token, while truncation shortens text that exceeds the predefined limit. A common value for the maximum sequence length in BERT-based models is 512 tokens, as it matches the architecture's capacity (HuggingFace, 2025).

To turn tokens into numerical input, embeddings are used. Earlier models like Word2Vec created static embeddings. But modern transformer models generate contextual embeddings. This means the meaning of a word adapts based on the surrounding words. Once the words are converted into token embeddings, their positions must also be encoded. This is because Transformers process all words at the same time and need positional information to understand the word order. The model uses a mechanism called self-attention. This allows the model to consider all words when processing each token. It helps the model understand context and resolve references, like pronouns. Multi-Head Attention lets the model focus on different parts of the text in parallel. This improves comprehension. Additionally, feedforward layers and normalization help improve the representation and make training more stable (HuggingFace, 2025).

IndoBERT can be fine-tuned for various tasks like homicide article classification or information extraction. Like most state-of-the-art models, it is flexible but relies on supervised learning to adapt to different tasks. This means it requires a labeled training dataset to learn. However, acquiring high-quality annotations is expensive and time-consuming, especially in low-resource languages. Since the model is pretrained on Indonesian, we can use transfer learning to benefit from its understanding of the language. Therefore, the model needs a relatively small annotated dataset compared to building a model from scratch or using a transformer not specialized in Indonesian(Pongpaichet et al., 2024; Silalahi et al., 2022).

The challenge of establishing an annotated dataset has led to increased interest in semi-supervised methods, which use a mix of labeled and unlabeled data to improve performance while reducing labeling costs. Nevertheless, supervised learning remains the most widely used approach due to its high accuracy and reliability (Chen et al., 2020; Subowo et al., 2025).

Another limitation of IndoBERT is its long execution time, making it computationally intensive. It requires GPUs for both training and inference. Its performance also depends on the quality and variety of the training data. Even though it needs fewer labeled examples than other models, fine-tuning still needs a good amount of data to avoid the risk of overfitting with small datasets.

The model's architecture is complicated and opaque. IndoBERT has multiple layers. This includes 12 transformer layers (also known as encoder layers), each with 768 hidden units, 12 attention heads, and a feed-forward layer size of 3,072 . For each task, only the final layers are changed. The first layers are kept the same for basic language understanding. This makes it challenging to fine-tune and understand the decision-making process for classification. The model training involves several key parameters. The learning rate controls how fast the model updates its weights. The Batch size decides how many samples are processed at once. Weight decay helps prevent overfitting by regularising the model. Models are trained over multiple epochs. This means the model goes through the training data several times. After each epoch, the model's performance is checked. Early stopping is used to stop training when the model stops improving on the validation set. This avoids overfitting. F1-score-based model selection is commonly used to evaluate performance and pick the best model without overtraining (HuggingFace, 2025).

2.5 Evolving NLP Applications in Crime and Journalism

Overall, there has been a shift from early systems focused on summarizing text and sentiment analysis to more complex tasks, such as topic modeling and detailed information extraction. This is all possible thanks to the rise of machine learning and deep learning to better capture the contextual meaning of unstructured data. In journalism, NLP has been used to cluster news articles and detect misinformation. In crime research, NLP is now being explored to gain early insights into criminal activity and public reactions (Alatrista-Salas et al., 2020). It is especially useful for tracking crime trends over time using large sets of news articles. Studies have used NLP to detect sudden rises in crimes such as cybercrime, fraud, and drug trafficking. Some researchers have even built crime event datasets from news reports when official data was missing, especially in places with low transparency.

Existing research uses a range of ML models—such as SVM, Naïve Bayes, and KNN—to classify articles as crime-related or not, and then into specific crime categories. There has been growing interest in implementing deep learning models for multi-class classification of crime topics (Bifari et al., 2024; Ku and Leroy, 2014; Norouzi, 2022). Across many studies, SVM and Random Forest have performed best for multi-class classification of crime-related articles. This is because they are effective at handling large volumes of imbalanced data, which is common in crime datasets(Alatrista-Salas et al., 2020; Bifari et al., 2024; Dharviyanti and Wilantika, 2024).

2.6 Addressing an Imbalanced Data Set

Based on different studies, a minimum of 5,000 articles was often needed for good performance. Most studies used over 10,000 articles. This was especially important because crime-related articles are a small part of all news articles (Alatrista-Salas et al., 2020; Bifari et al., 2024; Norouzi, 2022). Homicide-related articles are an even smaller portion. The general recommendation is to have at least 500 to 1,000 homicide-related articles to serve as positive examples.

Alternatively, many studies have used solutions to deal with the class imbalance between crime and non-crime articles. This helps reduce annotation effort. For example, in several studies, articles were first filtered with RegEx. This was combined with rule-based models and machine learning to create hybrid models. These models combine the generalization power of machine learning with specific rules to improve accuracy. (Norouzi, 2022; Sedik and Romadhony, 2023; Subowo et al., 2025)

Another common method is undersampling or oversampling. These change the annotated dataset. Undersampling reduces the number of majority class examples (non-homicide articles) during training. Oversampling increases the number of minority class examples (homicide articles) by duplicating or augmenting them.

Undersampling is helpful because it creates a smaller dataset and speeds up training. It also balances the data so the model can focus on both classes equally and reduces bias toward the majority class. However, removing too many non-homicide examples can lead to information loss. The model may miss important variations in the non-homicide class. This makes undersampling less ideal for small datasets. (Arefeen et al., 2019)

On the other hand, oversampling is great because it keeps all the original information and balances the classes. It works better for smaller datasets and helps the model generalize. But it comes with risks. If rare examples are duplicated too much, the model may overfit. That means it memorizes examples instead of learning general patterns. Also, training time is longer since the dataset is larger. Finally, using too much synthetic data can introduce noise or unrealistic samples. (Zhang et al., 2024)

The main issue with over- or undersampling is the need to reduce class imbalance without eliminating it completely. The model should be exposed to data that reflects the true distribution of real-world news articles. An overly balanced dataset can introduce bias and reduce performance on real data. It is important that the model learns to make predictions in the presence of natural imbalance between homicide and non-homicide related cases. Therefore, the class ratio should not be adjusted to exactly 1:1. Studies have shown that maintaining a moderate imbalance, such as a 2:1 or 3:1 ratio, can lead to better generalization. (Aymaz, 2025)

Instead of altering the training data, another approach is to create a weighted loss function. This method assigns a higher loss to misclassifying the minority class. This means that the custom loss function penalizes the model more for false negatives, which misclassify a homicide article as non-homicide related. To establish class weights for the loss function, the weight for each class is set as the inverse of its frequency. This helps the model focus on the minority class.

$$w_i = \frac{N}{n_i}$$

where: w_i is the weight for class i,
N is the total number of samples in the training set,
n_i is the number of samples belonging to class i.

Using weighted classes is helpful because it saves memory and avoids duplicating data. But if the weights are too imbalanced, the model can become unstable or overfit the minority class. This is a risk when the class imbalance is large, like with homicide and non-homicide articles. If the ratio is more than 20 times, the model may focus too much on the minority class. This leads to poor generalization and too many errors on the majority class, which lowers overall accuracy. One solution is to use a different class weight formula that isn't directly based on the inverse of class frequency. Another is to combine class weights with undersampling or oversampling. These steps can help stabilize training and improve results (Ahmad et al., 2018; Bose and Sarkar, 2024).

2.7 Trade-offs Between Simplicity and Contextual Intelligence

In sum, RegEx-based classification offers a fast, transparent, and resource-efficient solution, particularly useful for early filtering in low-resource settings. However, its rigid rule structure, poor generalization, and lack of contextual understanding limit its scalability and accuracy, especially with morphologically rich languages like Indonesian.

In contrast, embedding-based models like IndoBERT capture linguistic nuance and context, enabling higher classification accuracy and robustness across varied expressions. Yet, they demand labeled data, significant computational resources, and careful fine-tuning.

Ultimately, the choice depends on the use case: RegEx suits simple, interpretable filtering tasks, while embedding-based models are better suited for large-scale, nuanced classification. A hybrid approach—using RegEx for initial filtering followed by machine learning—may offer a balanced trade-off in performance and efficiency.

Table 1: *Comparison between Regex-based and ML-based Models for Article Classification and Information Extraction*

Aspect	**Regex-Based Model**	**ML-Based Model (e.g., BERT)**
Learning Type	Unsupervised (rule-based)	Supervised (requires labeled data)
Computational Power	Very low	High (requires GPU/TPU)
Execution Time	Fast (simple pattern matching)	Slower (due to deep model inference)
Transparency / Interpretability	Very high (rules are human-readable)	Low (black-box decision making)
Expert Knowledge Needed	High (rules must be manually crafted)	Moderate (needed for annotation more than model mechanics)
Ease of Adapting / Fine-tuning	Low (requires manual rule editing)	High (can retrain or fine-tune on new data)
Generalization Ability	Poor (limited to explicit patterns)	Strong (handles variation and unseen cases)
Scalability	Limited (rules don't scale well)	Scalable (suitable for large, complex datasets)
Robustness to Language Variance	Low (fails with rephrasing, synonyms, or typos)	High (context-aware and robust to variation)
Maintenance Over Time	Tedious (frequent manual updates)	Easier (retrain with updated data)

2.8 NER Information Extraction

NER is an NLP classification task that finds and classifies important words or phrases in the text. These are called "named entities". Many studies use NER models to identify key information like people, places, and dates. The model works by reading text and checking each word or group of words. It uses rules or machine learning to decide what kind of entity it is. Most modern systems use deep learning and train a transformer-based model pre-trained on the target language for the specific task. This helps reduce the need for expert knowledge used in traditional rule-based systems (Alatrista-Salas et al., 2020; Bose and Sarkar, 2024; Dharviyanti and Wilantika, 2024). The use of pre-trained models like BERT or IndoBERT removes the need for dictionaries or expert rules. These models can learn contextual patterns and entity types from large text data. This makes classification faster and easier without heavy work on resources (Ma et al., 2021; Pongpaichet et al., 2024; Silalahi et al., 2022; Zhou et al., 2022).

To build an NER model, the transformer model is initialized with a list of predefined lists of named entities. During training, the model sees text with correct entity labels. Over time, it learns to find new entities by itself. Most models focus on key information such as people (suspect, victim, witness), and their attributes like age, gender, or ethnicity. Other common points include time, place of the murder, weapon, and motive. These are the typical homicide attributes extracted across different models.

For example, the model takes a sentence and labels each word with either an entity or "O" (no entity):

"On Monday night, at a boarding house in Yogyakarta, Andi Saputra stabbed and killed Rina Marlina around 10 PM due to jealousy after discovering she was planning to leave him."

In this sentence, the model can extract:

- Name of perpetrator: Andi Saputra
- Name of victim: Rina Marlina
- Time: Monday night, around 10 PM
- Place: Boarding house in Yogyakarta
- Motive: Jealousy because she was planning to leave him

All other words would be tagged as "O", meaning they do not correspond to the predefined name entities.

One of the greatest challenges of using the model is preparing the labeled dataset to be compatible with IndoBERT's built-in training API. This process requires converting the standard labeled data into BIO format, where each token is assigned a specific label.

- **B-<ENTITY>** = Beginning of an entity
- **I-<ENTITY>** = Inside an entity (continuation of the same entity)
- **O** = Outside any entity (not part of a named entity)

Table 2: This table presents the token-level annotation of a sample sentence using the BIO (Beginning, Inside, Outside) scheme for NER. Each token is labeled to indicate whether it marks the beginning (B-) or is inside (I-) a named entity, or falls outside (O) of any entity. The entity types correspond to crime-related categories such as victim and suspect names, place, time, motive, and weapon. The sentence is split into two subtables for clarity, showing how the BIO format captures entity boundaries and types for model training. This annotation format is essential for training NER models like IndoBERT to identify and classify entities within crime reports.

Token	**Label**
On	B-TIME
Monday	I-TIME
night	I-TIME
,	O
at	O
a	O
boarding	B-PLACE
house	I-PLACE
in	I-PLACE
Yogyakarta	I-PLACE
,	O
Andi	B-SUSPECT$_{NAME}$
Saputra	I-SUSPECT$_{NAME}$
stabbed	O
and	O
killed	O

(a) First half of the annotated sentence

Token	**Label**
Rina	B-VICTIM$_{NAME}$
Marlina	I-VICTIM$_{NAME}$
around	B-TIME
10	I-TIME
PM	I-TIME
due	O
to	O
jealousy	B-MOTIVE
after	O
discovering	O
she	O
was	O
planning	O
to	O
leave	O
him	O
.	O

(b) Second half of the annotated sentence

Establishing the correct BIO tagging function is very important. This allows you to simply annotate the data. The function needs to go through the process of identifying each named entity's exact location in the text. It must then tag the beginning and inside of the entity. Otherwise, doing this manually would be too time-consuming. This is very important since the training data needs to have the labels correctly aligned. That way, the model can correctly learn how to predict the entities. This is a particular challenge that NER deals with is correctly detecting the exact span of named entities. For example, capturing only the victim's first name instead of the full name is a problem, since the full name is important.

Another big limitation it experiences is the misclassification of entities. This is especially common with person names, where the victim and suspect may be confused. This issue is particularly

prevalent in texts that use co-reference for the people involved, such as pronouns like "she" or aliases that often consist of initials or nicknames. One reason the NER model is challenged by this is that it processes each sentence independently and in parallel. So, it is hard for it to link a reference to the correct entity.

This issue can also be seen as some entities being missed, especially when they are rare. In order to properly learn through supervised learning, the model needs at least several hundred to a few thousand instances per entity type. This allows it to learn meaningful patterns because it needs many examples to learn how to generalize. Given the high demand for many examples of entity instances, NER could benefit from data augmentation. This is because annotating each article by hand and structuring the information to have a labeled dataset is very time-consuming. As such, data augmentation is very beneficial to increase the dataset size for training and help the model learn better. However, excessive data augmentation of the existing data is also counterproductive because it can lead to overfitting on artificial variations. This means the model might learn patterns that do not generalize well to real-world data. Moreover, augmented data may introduce noise or unrealistic examples, which can confuse the model and reduce its overall accuracy (Ma et al., 2021; Hashimoto et al., 2024; Chen et al., 2021; Yu et al., 2023; Zhou et al., 2022).

Therefore, it is important to balance augmentation carefully to improve diversity without compromising data quality. The recommended range to increase the dataset is by 2 to 5 times. Anything greater than 10 times would be considered excessive data augmentation, which could cause the model to overfit to synthetic or repetitive patterns and reduce generalization.

2.9 RE Information Extraction

RE models are designed to identify and classify semantic relationships between entities within text. For example, they detect connections between a suspect and a crime, or the time an event occurred. Typically, these models take as input the entities detected by NER as well and use that to analyse the text (Xu and Zhang, 2023).

The model considers all pairs (or sometimes sets) of entities to examine possible relations. The RE model converts the text and entity pairs into a format suitable for machine learning. Using deep learning models like IndoBERT, the input is tokenized and embedded into numerical vectors that capture the meaning of words and their context (Alt et al., 2019).

RE models analyze the entire sentence or surrounding context to understand how the entities relate. This helps distinguish different types of relationships, even if they are expressed differently. For example, in the sentence used above, the model can take the entity for the suspect's name and the victim's name. The RE model would define the relation between both entities as the triplet: *(Andi Saputra, KILLED, Rina Marlina).*

Modern RE approaches often use deep learning techniques, including transformer-based architectures. These methods capture complex dependencies in text and improve accuracy over traditional rule-based methods. One strength of RE models is their ability to generalize from training data. They

can detect relationships even in varied or unseen linguistic contexts (Alt et al., 2019; Xu and Zhang, 2023).

This reduces the need for extensive manual rule creation. It acquires a better contextual understanding of nuanced text. This is valuable for supporting downstream applications like question answering or knowledge graph construction.

However, challenges remain. RE models rely on supervised learning and require high-quality labeled datasets, which are costly and time-consuming to produce. Supervised learning uses fixed labels that must be predefined. In order to have a high number of instance examples for each relation, it needs a large dataset. In general, the model's performance depends on the accuracy of the NER model. It requires the entities to be defined as input before determining the relations between them.

Moreover, the model is extremely computationally expensive—more so than NER—since it evaluates all possible entity pair combinations per sentence. Additionally, RE can struggle with ambiguous or implicit relations that are not explicitly stated in the text. It is particularly challenged by overlapping or multiple relations between a single pair of entities. For example, if the suspect and victim also have a relationship such as siblings, then both the *sibling* relation and *killing* relation must be captured (Chen et al., 2023).

RE offers powerful capabilities for extracting structured insights from text, but its success depends heavily on careful annotation, computational resources, and robust NER. When done right, it provides essential depth to automated text understanding, especially in complex domains like crime reporting.

2.10 Related Work on Indonesian Crime Mining

For this study, a literature review was conducted on Indonesian news mining for crime data. The research utilized Google Scholar with the search query: ("crime" OR "criminal") AND ("Indonesia" OR "Indonesian") AND ("named entity recognition" OR "NER" OR "relation extraction" OR "RE"). The PRISMA research technique guided the careful selection of relevant papers. Only peer-reviewed articles published within the past five years were included. Six relevant studies were selected, each examining diverse datasets and employing various machine learning techniques to extract structured data from textual sources.

Sulastri et al. (2023) investigated gender bias in Indonesian online crime news by applying word embedding techniques (Word2Vec CBOW) on 1,560 crime-labeled news summaries from Detik.com (Sulastri et al., 2023). Using PCA visualization and gender ratio analyses, the study revealed that male-related terms are disproportionately associated with violent crime, exposing stereotype reinforcement in news media. This work highlights the societal implications of linguistic bias and underscores the necessity for bias mitigation in natural language processing (NLP) applications dealing with crime reporting. Although the study innovatively employs word embeddings to uncover implicit gender bias and raises awareness about media language shaping public perception and policy, its focus remains mainly on word embeddings without extending to contextual embeddings

or deeper semantic analysis. Additionally, the moderate dataset size suggests that expansion could improve generalizability.

Subowo et al. (2025) developed a specialized annotated corpus and an expert-validated named entity recognition (NER) model for identifying legal entities such as articles, laws, and sanctions in 450 Indonesian Supreme Court corruption decisions (Subowo et al., 2025). Their model leveraged IndoBERT with a conditional random field (CRF) layer, achieving an impressive F1 score of 0.923, outperforming BiLSTM-CRF and CRF baselines. The corpus, annotated with 12,000 entities in IOB format and reviewed by legal experts, represents a critical resource for Indonesian legal NLP. Despite these strengths, the study's focus on corruption rulings limits transferability to other legal domains, and the relatively small corpus size compared to general-domain datasets poses a limitation.

Sedik and Romadhony (2023) combined NER with sentence-level classification to extract crime types, locations, and dates from 1,963 Indonesian crime news articles (Sedik and Romadhony, 2023). Their pipeline utilized SpaCy NER for temporal and locational entities alongside support vector machines (SVM) for crime type classification, yielding F1 scores up to 0.95 for dates and 0.91 for crime types. This hybrid approach outperformed previous part-of-speech (POS) and dependency-based extraction methods. While the multi-component system improves precision and provides comprehensive evaluation on diverse entity types relevant to crime reporting, it relies on a relatively small annotated sentence set of 357 sentences and does not incorporate recent transformer-based models that could potentially enhance performance.

Yustina et al. (2024) explored rule-based extraction methods augmented with dependency parsing and ontology-based classification on a small dataset comprising approximately 40 news articles and 533 sentences (Yustina et al., 2024). Their system achieved a moderate overall F1 score of 0.607, with strong performance for crime type extraction (F1 score of 0.87.4), but struggled with other entity types due to limitations in language tools. The tailored linguistic rules and ontology integration provide interpretability and demonstrate the feasibility of rule-based systems in low-resource Indonesian NLP scenarios. However, the small dataset limits generalizability, and lower recall and F1 scores on certain entities highlight challenges inherent to rule-based approaches in complex language contexts.

Ulfatriyani et al. (2020) applied term frequency-inverse document frequency (TF-IDF) techniques on 27,639 preprocessed Indonesian tweets mentioning "penipuan" (fraud) to identify common modus operandi in fraud crimes (Ulfatriyani et al., 2020). Visualization using word clouds and histograms revealed frequently discussed terms, suggesting social media as a valuable source for crime pattern analysis. This study demonstrates the applicability of simple statistical text mining techniques on large-scale social media data and highlights social media as an alternative data source for crime analysis in Indonesia. Nonetheless, it lacks supervised learning or validation metrics such as F1 scores, and the analysis is limited to term frequency without contextual understanding.

Mantoro et al. (2022) collected tweets and Facebook posts covering ten crime categories and used multi-classifier models including logistic regression, SVM, naïve Bayes, and decision trees, to classify posts and generate a crime index (Mantoro et al., 2022), The best models achieved F1 scores up to 1.00 (SVM) and overall accuracy near 90%, demonstrating robust crime trend detection via social

media text mining. This comprehensive multi-classifier approach with high accuracy presents an innovative creation of a dynamic crime index based on social media monitoring. However, the study focuses primarily on classifier performance, lacking a detailed analysis of linguistic features driving classification, and may suffer from potential data bias due to social media user demographics.

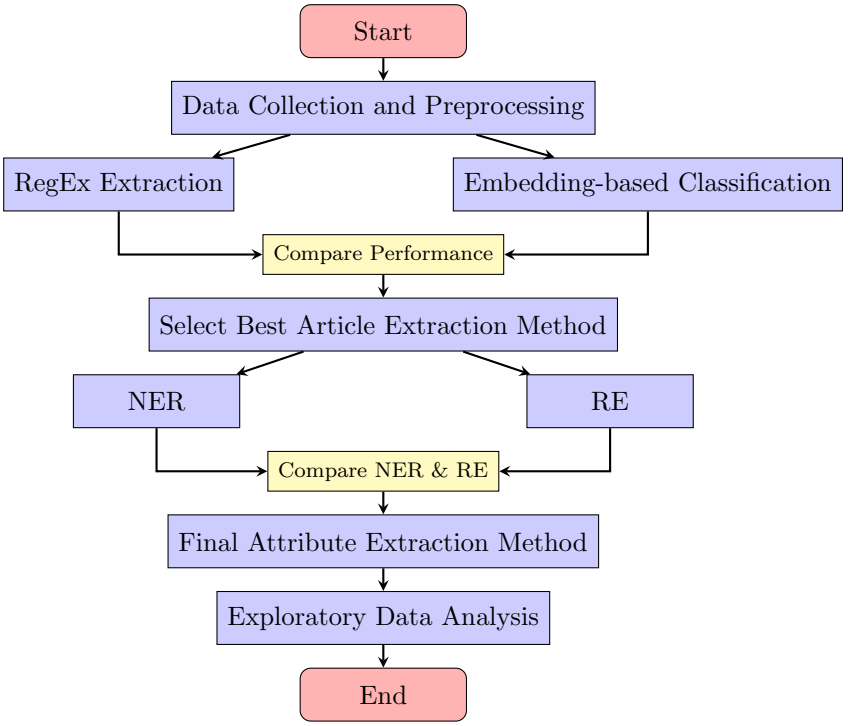
Collectively, these studies illustrate the rapid development of NLP applications in Indonesian crime and legal domains. Transformer-based approaches, particularly IndoBERT, now dominate tasks such as NER in legal texts, while classical machine learning and rule-based methods continue to contribute to information extraction from news and social media. Social media mining emerges as a complementary data source, albeit with challenges related to bias and noise. Research gaps remain, including the need to address gender and social biases in crime reporting through more advanced debiasing methods integrated into downstream NLP tasks, the expansion of expert-validated annotated corpora for diverse Indonesian legal and crime domains to improve model robustness, and the combination of rule-based ontologies with deep learning techniques to enhance interpretability without sacrificing accuracy. Furthermore, broader incorporation of transformer-based models beyond classification and NER—such as relation extraction and event detection—remains underexplored.

3 Methodology

3.1 Article Classification Methodology

This subsection describes the full implementation of the Indonesian News Mining Pipeline, which is illustrated by Figure 1. The pipeline consists of a binary classifier that determines whether a news article is homicide-related or not. Two contrasting approaches were employed: a rule-based method utilizing RegEx and a machine learning-based method utilizing contextual embeddings. This step was essential to ensure that only semantically relevant news articles were passed on for further attribute extraction.

Figure 1: The flowchart below illustrates the overall workflow of the article extraction and analysis process. It begins with data collection and preprocessing, followed by two parallel extraction methods: RegEx-based and embedding-based classification. The performance of these methods is compared to select the best article extraction approach. Subsequently, the selected articles undergo further processing through NER and RE. These outputs are then compared to determine the optimal final attribute extraction method.



3.1.1 Regex Classification Methodology

As a baseline, a rule-based keyword matching method usingRegEx was applied. The initial keyword list was minimal. The Indonesian team only provided two terms: *pembunuhan* (murder) and *dibunuh* (killed). According to them, these were the most precise and widely used terms, as Indonesian has no other direct synonyms for murder or homicide. Using only these words ensured that only homicide-related articles were extracted, reducing the chance of irrelevant results. However, this minimal list missed common variations such as *pembunuh* (murderer) or *membunuh* (to kill).

To address this, the keyword list was expanded through manual inspection of news articles and linguistic intuition. Articles were translated to English to identify relevant homicide-related phrases, which were then back-translated into Indonesian. This process yielded a total of eight keywords, including morphological variants and synonymous expressions.

Since the RegEx classifier serves as our baseline model, minimizing the false positive rate was prioritized over minimizing the false negative rate. In other words, precision was considered more important than recall, as the goal was to ensure that only clearly relevant articles were labeled as homicide-related. Each additional keyword was included in the final list only if its contribution did not significantly increase the false positive rate compared to the initial keyword set. This conservative selection strategy aimed to maintain high precision while slightly broadening coverage.

```
HOMICIDE_KEYWORDS = [
    r"\bpembunuh\w*\b",          # killer / murder / related nouns
    r"\bdibunuh\w*\b",           # was killed (passive)
    r"\bterbunuh\w*\b",          # killed (passive)
    r"\bpembantaian\w*\b",       # massacre / slaughter
    r"\bmenghabisi\s+nyawa\b"    # phrase: to take a life
]
```

Figure 1: Regex patterns used to identify homicide-related articles. The keywords include variations of common Indonesian terms such as `pembunuh` (murderer), `dibunuh` (was killed), and `pembantaian` (massacre). The RegEx allow for flexible matching of different word forms by using suffix wildcards (e.g., `[a-z]*`) and word boundaries (`\b`). Additionally, expressions such as `menghabisi nyawa` (to take someone's life) are matched as exact phrases using whitespace-aware patterns. This implementation balances coverage and precision when filtering relevant articles.

The model starts by cleaning the text. This involves converting all letters to lowercase and removing punctuation to normalize expressions. Python's re module was used to search for at least one keyword match in the article title or body text. The final regular expressions were designed to capture both exact matches and common suffix-based variations using wildcard and word-boundary logic. If a match was found, the article was labeled as relevant (1); otherwise, it was labeled as irrelevant (0). This binary classification served as a first-pass filter. Overall, this step ensured that only potentially relevant articles moved on to further processing.

3.1.2 Embedded-Based Classification Methodology

To overcome the limitations of keyword-based filtering, a supervised machine learning approach was implemented. This method uses contextual sentence embeddings from the IndoBERT language model to classify articles based on their semantic content.

Since the embedding-based model requires supervised learning, a subset of the scraped corpus was manually labeled. The annotated dataset consisted of approximately 10,189 articles. Each article was independently inspected and labeled with a binary class: homicide-related (1) or not homicide-related (0). Classification was based on whether the article reported an intentional killing involving a victim and a perpetrator. Reports of suicides, accidents, or non-lethal violence were labeled negative unless clearly linked to unlawful death.

Out of 10,189 labeled articles, only 396 (3.89%) were homicide-related. This shows a severe class imbalance, with homicide reports representing a small minority. To ensure reliable evaluation, the dataset was split into training (70%), validation (15%), and test (15%) sets using stratified sampling. This preserved the original class distribution in each subset.

This text was then broken down into smaller parts using a tool called AutoTokenizer. This tokenizer

came from the HuggingFace Transformers library and was specifically made for the IndoBERT model. The maximum sequence length was set to 512 tokens, with padding and truncation enabled. This means that the text is cut off or filled up with special tokens when it is too short. This made the input compatible with the model and ensured efficient training and evaluation.

For the actual model, AutoModelForSequenceClassification was used. This means the model takes the tokenized text and predicts whether the article is homicide-related or not. It does this by assigning a probability to each class using a function called softmax, which helps the model decide between the two options.

The model was trained using the Trainer class from HuggingFace Transformers. Some key training parameters were set to be standardized across all models:

- A learning rate of `2e-5`, which controls how quickly the model updates its knowledge.
- A batch size of `8`, meaning the model looked at 8 articles at a time before adjusting itself.
- The model was trained for 3 epochs, where it went through all the data 3 times.
- A small weight decay of `0.01` was used to prevent the model from overfitting (getting too focused on the training data).

After each training epoch, the model was evaluated to see how well performed on the classification task. The best version of the model was chosen based on the F1-score. Five distinct embedding-based classification models were produced:

- The basic_EB_indo_bert_model trained on the data as-is, it does not provide a method to address class imbalance in the news articles.
- The weight_EB_indo_bert_model used weighted classes, penalizing misclassification of homicide articles more than non-homicide ones. Class weights were calculated from class frequencies in the training dataset. Thus, the computation assigns a higher weight to the minority class (homicide) and a lower weight to the majority class (non-homicide). These weights are integrated into the loss function by using a weighted cross-entropy loss during model training, which penalizes misclassification of minority class samples more heavily. The algorithm establishes a custom loss function to help the model learn balanced decision boundaries despite the skewed data distribution
- The ratio_EB_indo_bert_model uses another approach to handle class imbalance by applying random undersampling to reduce majority class examples. For the training set, the minority class samples (homicide articles) are kept intact, while the majority class (non-homicide) samples are undersampled at varying ratios. For each ratio, a subset of the majority class is randomly sampled without replacement to match the target ratio, and these samples are combined with all minority class instances to create a balanced training dataset. Different ratios from 1:0 to 1:9 (homicide to non-homicide) were tested to find the best balance.

- The weight_ratio_EB_Indo_bert_model methodology integrates both the weighted loss and the undersampling techniques to address class imbalance and optimize model performance. Specifically, this approach calculates the class weighting from the training dataset before undersampling is within the loss function, assigning higher penalty weights to the minority class (homicide). Subsequently, it alters the training dataset by systematic undersampling of the majority class (non-homicide), thus selectively reducing the number of majority class samples at varying ratios from 1:0 to 1:9
- The best performing embedded-based model is further optimized with hyperparameter tuning on learning rate, batch size, epochs, and weight decay. Learning rates between 1e-5 and 5e-5, batch sizes of 8, 16, and 32, epochs between 3 and 5, and weight decay from 0 to 0.1 were tested using grid search. This method determines the best parameters to optimize the model's performance.

In summary, the embedding-based classification methodology leveraged IndoBERT to capture the semantic context of news articles and effectively detect homicide-related content. By experimenting with various imbalance-handling strategies and fine-tuning hyperparameters, the approach aimed to optimize model performance despite the rarity of homicide cases in the dataset.

3.2 Infomation Extraction Methodology

Extracting key characteristics related to homicide incidents from news articles is a crucial step to understanding the circumstances and actors involved in each case. This section outlines the methods used to identify named entities relevant to homicides and the relationships between them. NER and RE were implemented, leveraging both dependency parsing and Transformer-based models. Both models were trained on IndoBERT to utilize the pre-trained BERT-based language model's ability to capture nuances in Indonesian syntax and semantics. This allowed effective recognition of entities and the relationships connecting them.

3.2.1 NER Methodology

The NER system was designed to identify eight entity types relevant to the characterization of homicide cases:

- **VICTIM_NAME:** Full name of the harmed or killed individual.
- **VICTIM_AGE:** Age of the victim at the time of the incident.
- **SUSPECT_NAME:** Full name of the alleged perpetrator.
- **SUSPECT_AGE:** Age of the suspect.
- **PLACE:** Specific location where the incident occurred.

- **TIME:** Time or date of the incident.
- **WEAPON:** Object or means used in the homicide (e.g., firearm, knife).
- **MOTIVE:** Suspected reason behind the crime (e.g., robbery, jealousy).

To accomplish this, the NER model was fine-tuned on a labeled dataset. A corpus of Indonesian news articles related to homicide cases was manually annotated using a custom annotation schema that included eight preselected entity types. In total, 396 articles were annotated. The resulting dataset was randomly split into training, validation, and test sets using a 70:15:15 ratio. While the original dataset was sufficient to develop a prototype model, data augmentation was applied to enable a more advanced version. Specifically, synonym replacement and entity swapping techniques were used to expand the labeled dataset, resulting in a training corpus of approximately 1,200 samples. The data augmentation algorithm is explain in depth in subsection Section 3.3.

Before training IndoBERT, the text was cleaned using a custom *prepare_dataset* function. This step is necessary because IndoBERT requires precise alignment of BIO labels. The Algorithms clean the textual data by removing all punctuation, converting all letters to lowercase, and normalizing the text for tokenization. Additionally, the secondary function, *expand_named_entities*, handles entities with multi-token names by expanding them into individual tokens. This improves the model's ability to recognize partial mentions, common in news articles where a person is not always referred to by their full name.

The *bio_tagging_for_text* function uses the Hugging Face tokenization to encode entities into a BIO tagging scheme at the word level, labeling each token as B-ENTITY_TYPE (beginning), I-ENTITY_TYPE (inside), or O (outside any entity). This facilitates token classification during training. The custom matching algorithm aligns annotated entity spans with tokens in the text, flagging any entities missed during tagging for review.

Once the data is in the appropriate BIO format, the IndoBERT tokenizer splits the text into subword tokens. Tokens exceeding the maximum sequence length of 512 are truncated, and shorter sequences are padded to maintain batch consistency. The processed tokens and aligned BIO labels are then converted into PyTorch datasets compatible with the Hugging Face Trainer API.

At this stage, the IndoBERT model is initialized, trained, and fine-tuned using the AdamW optimizer with a learning rate of 5e-5 over 5 epochs, batch size 8, and weight decay of 0.01. Early stopping was applied based on validation set performance.

After training completes, an evaluation strategy determines which model configuration and tokenizer produced the best-performing NER. The model is assessed using the seqeval library, which computes precision, recall, and F1-score for sequence labeling tasks. Evaluation was based on validation set performance, and the best-performing model was saved for final testing.

Besides the basic NER model, a model with a weighted loss function was developed. This was important due to severe class imbalance. Class weights were derived from frequency distributions in the training data and applied to the loss function to ensure rarer entity tokens were given a higher penalty when misclassified.

The final NER model is derived from a grid search which explores hyperparameter combinations (learning rate, batch size, epochs, and weight decay). A grid search was conducted over learning rates {1e-5, 2e-5, 3e-5, 5e-5}, batch sizes {8, 16, 32}, weight decay values {0, 0.01, 0.05, 0.1}, and training epochs {3, 4, 5}. The best configuration was determined using validation F1-score and early stopping, which stopped the training if no improvement in F1 for 2 evaluation rounds.

In summary, the NER methodology combined manual annotation, strategic preprocessing, and model fine-tuning using IndoBERT to extract key homicide-related entities. Class imbalance was addressed through weighted loss, and performance was optimized via grid search and early stopping. This systematic approach resulted in a robust model capable of identifying diverse entity types relevant to homicide case characterization.

3.2.2 RE Methodology

This section describes the RE component developed to identify relationships between pairs of named entities previously extracted through NER. This pipeline focused on extracting seven predefined relations (see Theoretical Background).

The RE model implementation used dependency-based heuristics to guide relation prediction, leveraging syntactic structure between entities. Entity pairs were embedded into the input sentence using special tokens: the head entity was marked with `[HEAD]` and `[/HEAD]`, and the tail entity with `[TAIL]` and `[/TAIL]`. This format helped the model focus on the relevant entity pair during encoding.

Input sentences were tokenized using the IndoBERT tokenizer. Each example was truncated or padded to a maximum sequence length of 512 tokens to ensure consistency in batch processing. Entity pairs were assigned a label corresponding to one of the relation classes or to a "NO_RELATION" class when no meaningful link was identified.

The IndoBERT model was fine-tuned using supervised learning. Training was performed over three epochs with a batch size of 16 and a learning rate of 5e-5 using the AdamW optimizer. A manual training loop was implemented to iterate over the training data, compute cross-entropy loss, and backpropagate gradients to update model weights.

Model evaluation was based on the macro-averaged F1-score, with comparisons performed across relation classes. The annotated dataset created for NER was extended to include relation labels, allowing joint usage for both entity and relation extraction tasks.

The following Information Extraction model is RE. This model aims to identify semantic relationships between pairs of entities within text. This step is important to extract structured information from unstructured text data. The RE model focuses on extracting the following seven relations:

- **KILLED:** Connects SUSPECT_NAME to VICTIM_NAME, indicating the suspect allegedly killed the victim.

- **AT_LOCATION:** Connects VICTIM_NAME or SUSPECT_NAME to PLACE, specifying where the incident took place.
- **AT_TIME:** Connects VICTIM_NAME or SUSPECT_NAME to TIME, indicating when the incident occurred.
- **HAS_AGE:** Connects VICTIM_NAME or SUSPECT_NAME to their respective age (VICTIM_AGE or SUSPECT_AGE).
- **USED_WEAPON:** Connects SUSPECT_NAME to WEAPON, specifying the object or method used in the crime.
- **HAS_MOTIVE:** Connects SUSPECT_NAME to MOTIVE, indicating the suspected reason behind the crime.
- **NO_RELATION:** Indicates no meaningful relation exists between the two entities.

The RE model implementation leverages dependency parsing to determine relationships between entities. This approach uses grammatical structure as a cue for relation prediction. Dependency paths effectively indicate whether two entities participate in a meaningful relation by examining their syntactic connection. This linguistic foundation is further supported by the pretrained BERT model.

The Transformer-based RE model exploits language-specific capabilities and powerful contextual embeddings. This is essential to capture nuanced relationship patterns in Indonesian news texts.

The relation extraction task requires supervised learning to train the IndoBERT model to identify relationships between entities. The annotated dataset created for the NER model was extended to label relationships between annotated entities. Each entity pair is explicitly marked in the text, with the head entity wrapped in special tokens [HEAD] and [/HEAD], and the tail entity wrapped in [TAIL] and [/TAIL]. This marking guides the Transformer model to focus on the relevant entity pair when encoding the input sentence.

The marked sentences are tokenized using the pretrained BERT tokenizer. Inputs are truncated or padded to a fixed length of 512 tokens. Each entity pair is assigned a label corresponding to the predefined relation classes or "NO_RELATION" if no meaningful relation exists. During training, the model is fine-tuned using the AdamW optimizer with a learning rate of 5e-5 over three epochs. Training is done in batches of size 16 through a manual loop iterating over the training data. The process involves computing cross-entropy loss and backpropagating gradients, with model weights updated accordingly.Evaluation uses the macro-averaged F1-score. This metric measures precision and recall for each relation class to assess the model's ability to generalize across relation types.

3.3 Data Processing

The following section describes the data collection and preprocessing steps undertaken to build a comprehensive corpus for training and testing the news mining model. This involved implementing

a Python-based web scraper targeting Kompas.com, a reputable Indonesian news source selected for its accessible API and unrestricted article retrieval capabilities.

The scraper iteratively constructed URLs for Kompas' daily article indexes. By going through all dates and pages, the metadata could be fetched. To implement the scraper successfully, the requests library was used along with parsing the HTML structure via BeautifulSoup.

For each article, the script accesses the title, timestamp, and full text content. The collected articles are saved in JSON format as illustrated by Figure 2 , grouped by date, and stored locally in a structured folder. This scraping method allowed organized and reproducible downstream processing, which was crucial for the NLP tasks.

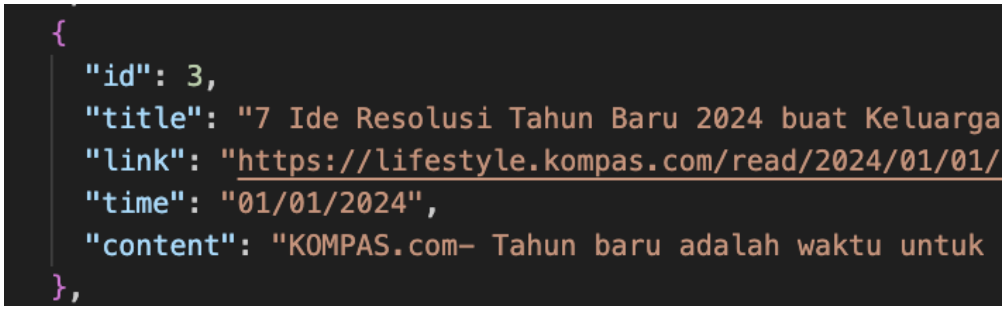


Figure 2: *The following code snippet demonstrates how the web scraper extracts news articles from an online source and organizes each article into a structured format. Each article is assigned a unique ID and includes fields such as title, link, date, and full content for further processing.*

Scraping was performed only on Kompas articles published in 2024. This years articles consisted of 279,678 articles. On average, each article had about 302 words and 20 sentences. Approximately 804 articles were released daily.

Kompas news articles were primarily written fully in Indonesian. During data exploration, some articles were found to include English, usually in short announcements for new smartphone launches or just a few words. For the purpose of this study, all articles are assumed to be completely in Indonesian. Close inspection of around 400 homicide-related articles confirms this assumption, as those articles were all in Indonesian, which is the relevant part of the study.

Since none of the articles were already annotated, manual annotation was necessary. Articles from January 1st to January 13th were selected to create an annotated dataset. This dataset consisted of 10,189 news articles, of which only 396 were homicide-related. This means only 3.38% of articles were homicide-related, showing a great class imbalance and highlighting the need for an annotated dataset. A new attribute was added to the JSON file to clearly label each article as 0 (not homicide-related) or 1 (homicide-related). An annotated news article sample is depicted in Figure 3 .

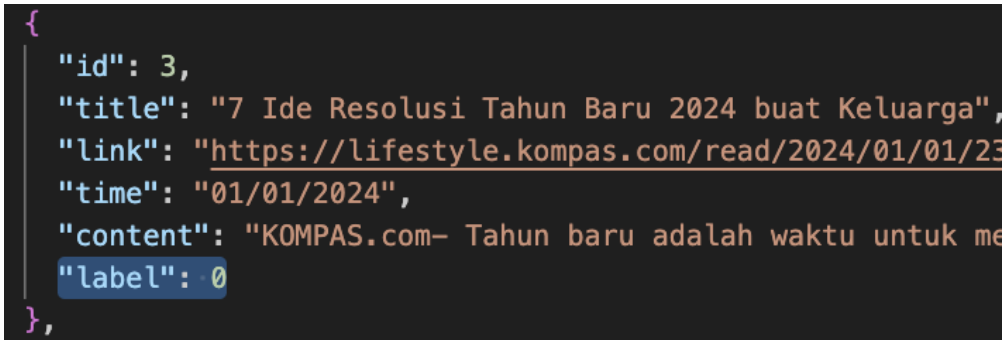


Figure 3: *This code snippet illustrates how a new feature, label, was manually added to each article during the annotation process. The label indicates whether the article is related to a homicide case (label = 1) or not (label = 0), supporting the binary classification task.*

After identifying around 400 homicide articles, entities and entity relations were annotated by going through each article. An annotated news homicide article sample is depicted in Figure 4 with named entities and relations labeled. For each identified entity, a label was assigned along with the exact text extract from the article corresponding to that entity. The text extract had to match exactly. If the entity boundaries were unclear, the model itself would determine where the full or partial entity occurred in the text during processing. For example, if the name "Anna Winter Lee" appeared, the full name was given as the entity exactly as written. The model needed to find all full or partial occurrences of the entity, even if only part of the name was mentioned multiple times. Entities were then extracted and noted exactly as written, except for motives. Motives are usually ambiguous and nuanced, rarely explicitly stated, so they cannot be easily extracted from the text.

```
{
    "id": 118,
    "title": "Kakek Tewas di Ponorogo Usai Rayakan Tahun Baru, Sak
    "link": "https://surabaya.kompas.com/read/2024/01/01/172655378
    "time": "01/01/2024",
    "content": "PONOROGO, KOMPAS.com-Satuan Reserse dan Kriminal P
    "entities": [
      {"label": "VICTIM_NAME", "text": "Ahmad Suyoto"},
      {"label": "VICTIM_AGE", "text": "52"},
      {"label": "PLACE", "text": "Dukuh Krajan, Desa Pulung, Kecam
      {"label": "TIME", "text": "Senin (1/1/2023)"},
      {"label": "MOTIVE", "text": "masalah pribadi"},
      {"label": "WEAPON", "text": "balok kayu"}
    ],
    "relations": [
      {"head": 0, "tail": 1, "label": "HAS_AGE"},
      {"head": 0, "tail": 2, "label": "AT_LOCATION"},
      {"head": 0, "tail": 3, "label": "AT_TIME"},
      {"head": 5, "tail": 0, "label": "KILLED"},
      {"head": 4, "tail": 5, "label": "HAS_MOTIVE"},
      {"head": 5, "tail": 6, "label": "USED_WEAPON"}
    ]
},
```


Figure 4: *This code snippet shows how a homicide-related news article is structured in the dataset after manual annotation. It includes the article's metadata, full text content, and a set of extracted named entities such as victim name, age, place, time, motive, and weapon. Additionally, the relations field links these entities to represent semantic relationships, such as who was killed, where the incident occurred, and what weapon was used.*

By listing the named entities, they were automatically numbered starting with index 0. This indexing also allowed the creation of relations between entities using their index. Compared to named entities, there are significantly fewer relations. Moreover, 400 homicide articles are simply insufficient to train a robust model; as such, we performed data augmentation. The data augmentation technique generates two new versions of each annotated article by randomly replacing entity texts with semantically similar alternatives from predefined lists. For example, the victim's name will be changed to another name. It ensures replacements occur both in the article content and the entity annotations, preserving entity-label integrity. Relations are updated accordingly to reflect the new entity indices. This method helps to increase training data without corrupting the structure.

Since the dataset of 400 homicide articles is too small to train a robust model, we applied data augmentation. The augmentation algorithm works by first scanning all annotated articles and creating a set of entity mentions for each entity type. For example, for the entity type **PLACE**, it collects all murder locations mentioned across the dataset into a single set; similarly, for **VICTIM_NAME**, it compiles all extracted victim names. Each entity mentioned is automatically

assigned an index, starting from 0, which also enables consistent tracking of relations between entities. Because there are far fewer relations than entities, indexing ensures that entity–relation mappings remain intact.

During augmentation, two new versions of each article are generated by randomly replacing entity mentions with alternatives drawn from these sets. For entities such as victim names, suspect names, or locations, replacements are performed using regular expressions to match and substitute all occurrences of the entity or its components in the article text, ensuring that partial mentions are also replaced correctly. Other entity types are replaced directly in the text. The entity annotations are updated in parallel, and relations are adjusted according to the new entity indices, preserving the structural consistency of the articles.

This method increases the training data size while preserving the integrity of entities, labels, and relations, ensuring that the augmented data remains consistent with the original annotation scheme.

Table 3: Entity and Relation Counts for Non-Augmented and Augmented Data

This table compares the counts of named entities and relation types extracted from the original (non-augmented) dataset and the augmented dataset. Augmentation significantly increases the number of detected entities and relations, enhancing dataset richness.

Category	**Non-Augmented Count**	**Augmented Count**
Entities		
VICTIM_NAME	407	1219
VICTIM_AGE	211	633
SUSPECT_NAME	390	1162
SUSPECT_AGE	184	552
PLACE	803	2405
TIME	585	1753
WEAPON	260	680
MOTIVE	201	603
Relations		
KILLED	316	918
AT_LOCATION	738	2186
AT_TIME	552	1634
HAS_AGE	329	965
USED_WEAPON	257	687
HAS_MOTIVE	217	615

Based on the distribution of entities and relations, an imbalance in the dataset is evident. These significant differences influence model performance. Among the entities, **PLACE** appears most frequently, increasing from 803 occurrences in the original dataset to 2405 after augmentation. This is because an article often specifies places in various ways—for example, initially naming the

country or city, and later referring to more specific locations such as a store or neighborhood. These are then labeled as distinct place entities.

TIME also occurs most frequently in the labeled dataset; in every article, time is extracted, often with explicit dates and times or vague references if exact times are unknown (e.g., "in the evening" or "in the morning"). Subsequently, the most frequent entities after augmentation are **VICTIM_NAME** and **SUSPECT_NAME**, both showing substantial growth in frequency. It is important to note that each named entity was only labeled once per subject. This means that, although names of victims and suspects appear multiple times within an article, only the first full occurrence was annotated, allowing the model to identify subsequent partial or full occurrences. Consequently, these entities are actually among the most frequent, even if this is not fully reflected in the labeled data.

Regarding relations, **AT_TIME** is the most common, rising from 738 to 2186 instances. The relations **KILLED** and **AT_TIME** also show notable increases in the augmented data. The least frequent relations, **USED_WEAPON** and **HAS_MOTIVE**, see only modest increases from 257 to 687.

The data collection and preprocessing pipeline successfully gathered a large and well-structured corpus from Kompas.com, enabling thorough NLP model training and evaluation. Manual annotation and subsequent data augmentation addressed the class imbalance and limited labeled data, enriching the dataset with varied named entities and relations. This robust dataset foundation is essential for effective model development and improved performance in extracting homicide-related information from news articles.

3.4 Experimentation

This section presents the experimental setup and evaluation methodology. The experiments aim to assess the effectiveness of classification and information extraction techniques for identifying homicide-related news articles and extracting key entities and their relations. The hardware and software environments, dataset splits, and evaluation metrics are described to ensure robustness, comparability, and reproducibility of results across tasks.

3.4.1 Experiment Setup

All experiments were conducted on a workstation equipped with an NVIDIA RTX 3090 GPU (24GB VRAM), Intel Core i9 CPU, and 64GB RAM. The models were implemented in Python 3.9 using the Hugging Face Transformers library (version 4.6) for embedding-based classification, the NER and RE models. PyTorch (version 1.3) was implemented as a deep learning framework. Additional packages included scikit-learn for metrics and data processing.

The dataset was divided into training (70%), validation (15%), and test (15%) sets using stratified sampling to preserve class distribution. For model evaluation, 5-fold cross-validation was performed

on the training set to tune hyperparameters and assess robustness. The final model performance was reported on the held-out test set to ensure unbiased results.

3.4.2 Evaluation Metrics

To evaluate the performance of the models, several standard evaluation metrics were used. These metrics accurately assess the models' ability for classification and information extraction tasks. Precision, Recall, and F1-score were the primary evaluation metrics.

Precision measures the proportion of true positive predictions among all positive predictions made by the model. It shows how many of the predicted positive cases were actually correct.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Where:

- **TP (True Positives)**: The number of correctly predicted positive instances (e.g., correctly identified homicide-related articles).
- **FP (False Positives)**: The number of instances incorrectly predicted as positive (e.g., non-homicide articles incorrectly labeled as homicide-related).

For classifying homicide articles, precision indicates how many articles classified as homicide-related were truly homicide-related. For information extraction, precision indicates whether a labeled token is actually an entity.

Recall (also called sensitivity) calculates the proportion of true positive cases correctly identified out of all actual positive cases. It reflects the model's ability to detect positive samples. For classification, recall shows if the model missed any homicide-related articles. For information extraction, recall shows if the model missed entities, leading to false labels or omissions of important information.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Where:

- **TP (True Positives)**: The number of correctly predicted positive instances (e.g., homicide-related articles correctly identified by the model).
- **FN (False Negatives)**: The number of actual positive instances that were incorrectly predicted as negative (e.g., homicide-related articles that the model failed to detect).

The F1-score is the harmonic mean of precision and recall. It provides a balanced metric accounting for both false positives and false negatives. This metric is especially useful when classes are imbalanced, where accuracy can be misleading. The F1-score was mainly used instead of accuracy because less than 4% of articles in the corpus are homicide-related.

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

In addition to the standard F1-score, two aggregation methods are commonly used for multi-class or imbalanced datasets: macro F1 and weighted F1. The macro F1-score calculates the F1 for each class independently and then averages them, giving equal weight to all classes regardless of their frequency. This can highlight poor performance on minority classes but may underrepresent performance on the majority class. In contrast, the weighted F1-score also computes F1 per class but weights each score by the number of true instances in that class, providing a more representative measure of overall model performance on imbalanced datasets. Since homicide articles and entity classes are highly imbalanced in our dataset, the weighted F1-score was primarily used to report results.

This means that if all articles were classified as not homicide-related, the model would still have around 96% accuracy, which could lead to missing all homicide articles. The same applies to information extraction, where most words are not entities, and the entities to extract form a minority class.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Where:

- **TP (True Positives)**: The number of correctly predicted positive instances (e.g., homicide-related articles correctly identified as such).
- **TN (True Negatives)**: The number of correctly predicted negative instances (e.g., non-homicide articles correctly identified as non-homicide).
- **FP (False Positives)**: The number of instances incorrectly predicted as positive (e.g., non-homicide articles wrongly classified as homicide-related).
- **FN (False Negatives)**: The number of instances incorrectly predicted as negative (e.g., homicide-related articles that the model failed to detect).

These metrics help evaluate the trade-off between avoiding false alarms (high precision) and finding as many relevant cases as possible (high recall). The F1-score was chosen because it is the standard

metric across many studies. Although the F2-score, which emphasizes recall over precision, could be useful for prioritizing not missing homicide-related articles or information, the F1-score was preferred for its balanced evaluation without favoring either metric.

For the NER and RE models, entity-level F1-scores were calculated to evaluate performance more precisely. The entity-level F1-score measures how accurately the model identifies and classifies individual entities and relations in the text. It considers both the correct boundaries (start and end) and the type of entities predicted. These F1-scores use exact matching criteria, where both entity boundaries must be correct to count as true positives. This strict evaluation ensures reliable information extraction for downstream analysis.

In the experiments, all metrics were calculated using the BERT standard evaluation libraries to ensure reproducibility and comparability with related work.

4 Results

This section presents the results of two core components of the pipeline: (1) the classification models used to identify homicide-related news articles, and (2) the information extraction models used to extract relevant homicide attributes from the classified articles.

For each task, model performance is reported using standard evaluation metrics such as precision, recall, and F1-score on the respective test sets. Homocide article classification results are presented for both the rule-based RegEx method and multiple IndoBERT-based variants. Homocide information extraction results include performance metrics for identifying key homicide elements such as perpetrator, victim, location, and time of homicide.

4.1 Article Classification Results

Table 4: Performance comparison between a RegEx-based model and various embedding-based classifiers for the classification task. The embedding-based models consistently outperform the RegEx model across all metrics. The best overall performance, in terms of F1-score, was achieved by the weighted embedding-based classifier with hyperparameter tuning. The highest recall was obtained by the combined weighted and undersampled classifier, though at the cost of increased false positives. Metrics shown include Accuracy, Precision, Recall, F1-score, False Positives (FP), False Negatives (FN), False Positive Rate (FPR), and False Negative Rate (FNR).

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**FP**	**FN**	**FPR**	**FNR**
RegEx Model	0.9696	0.7143	0.375	0.4918	5	15	0.0061	0.625
Simple Embedding-Based Classifier	0.9843	0.8158	0.775	0.7949	7	9	0.0072	0.225
Weighted Embedding-Based Classifier	0.9892	0.9143	0.8	0.8533	2	8	0.0031	0.0488
Undersampling Embedding-Based Classifier	0.9794	0.623	0.725	0.7342	10	11	0.0102	0.275
Weighted + Undersampling Embedding-Based Classifier	0.9745	0.619	0.9745	0.75	24	2	0.0245	0.0488
Weighted Embedding-Based Classifier with Hyperparameter Tuning	0.9901	0.9191	0.805	0.8606	1	7	0.0015	0.0427

Table 5: Effect of varying non-homicide to homicide article ratios on embedding-based classification performance using undersampling. Models were trained with increasing amounts of non-homicide data (from 0:1 up to 9:1 ratios) to assess the trade-off between precision and recall. The highest F1-score is observed at a 9:1 ratio, suggesting this ratio offers the best balance between capturing true homicide cases and limiting false positives.

Ratio (x:1)	**Accuracy**	**Precision**	**Recall**	**F1-score**
0	0.0391	0.0391	1	0.0750
1	0.9054	0.2659	0.8070	0.4885
2	0.9596	0.4884	0.7368	0.5874
3	0.9602	0.4947	0.8246	0.6184
4	0.9623	0.5104	0.8596	0.6405
5	0.9671	0.5529	0.8246	0.6620
6	0.9712	0.6230	0.6667	0.6441
7	0.9745	0.6190	0.9000	0.7310
8	0.9770	0.6210	0.9100	0.7330
9	0.9780	0.6280	0.9140	0.7430

The following six confusion matrices show the performance of the baseline RegEx model, the simple embedding-based model, the embedding-based model with class weighting, and the weighted embedding-based model with hyperparameter tuning. These models were chosen to compare both baselines and the two best-performing models. They demonstrate how performance improves step by step, while also highlighting different trade-offs between precision and recall. The remaining confusion matrix for the other 2 distinct classification models are included in the appendix.

Table 6: Confusion matrix of the RegEx model on the homicide article classification task. The true negatives are non-homicide articles, and the true positives are homicide articles.

Confusion Matrix: RegEx Model

	Predicted Negative	**Predicted Positive**
Actual Negative	973	5
Actual Positive	15	26

Table 7: Confusion matrix illustrating the classification performance of the simple embedding-based model on the homicide article detection task. The true negatives are non-homicide articles, and the true positives are homicide articles.

Confusion Matrix: Basic Embedding-Based Model

	Predicted Negative	**Predicted Positive**
Actual Negative	972	6
Actual Positive	9	32

Table 8: Confusion matrix demonstrating the classification performance of the weighted embedding-based model on the task. The true negatives are non-homicide articles, and the true positives are homicide articles.

Confusion Matrix: Weighted Embedding-Based Model

	Predicted Negative	**Predicted Positive**
Actual Negative	976	2
Actual Positive	8	33

Table 9: Confusion matrix demonstrating the classification performance of the under-sampling embedding-based model on the task. The true negatives are non-homicide articles, and the true positives are homicide articles.

Confusion Matrix: Undersampling Embedding-Based Model

	Predicted Negative	**Predicted Positive**
Actual Negative	968	10
Actual Positive	11	30

Table 10: Confusion matrix demonstrating the classification performance of the weighted + under-sampling embedding-based model on the task. The true negatives are non-homicide articles, and the true positives are homicide articles.

Confusion Matrix: Weighted + Under-sampling Embedding-Based Model

	Predicted Negative	**Predicted Positive**
Actual Negative	976	24
Actual Positive	2	39

Table 11: Confusion matrix of the weighted embedding-based model with hyperparameter tuning. The model achieved 976 true negatives and 33 true positives, with 2 false positives and 8 false negatives—indicating improved class distinction and a better precision-recall balance.

Confusion Matrix: Weighted Embedding-Based Model with Hyperparameter Tuning

	Predicted Negative	**Predicted Positive**
Actual Negative	976	1
Actual Positive	7	34

.

4.2 Information Extraction Results

4.2.1 NER Results

This subsection reports the performance of the NER model used to identify and classify entities such as persons, locations, time, motive, and weapons within homicide-related news articles. The results include precision, recall, and F1 scores for each entity type and demonstrate the model's ability to accurately extract key components for downstream relation extraction.

Table 12: NER performance by entity type for the homicide information extraction task. Metrics include precision, recall, and F1-score for each entity class, highlighting varying model effectiveness across different categories. While structured fields like `VICTIM_AGE` perform well, more abstract or context-dependent entities like `MOTIVE` and `TIME` show lower scores, indicating areas for further improvement. Macro and weighted averages are provided to reflect overall performance across all entity types.

Entity	**Precision**	**Recall**	**F1-Score**
VICTIM_NAME	0.3010	0.4247	0.3523
VICTIM_AGE	0.3750	0.3846	0.3797
SUSPECT_NAME	0.3519	0.2235	0.2734
SUSPECT_AGE	0.3750	0.4000	0.3871
PLACE	0.2254	0.2233	0.2243
TIME	0.2273	0.4375	0.2991
WEAPON	0.2273	0.1429	0.1754
MOTIVE	0.0000	0.0000	0.0000
Macro Avg.	**0.2603**	**0.2796**	**0.2614**
Weighted Avg.	**0.2622**	**0.2840**	**0.2645**

Table 13: NER performance by entity type for the homicide information extraction task with augmented data. Metrics include precision, recall, and F1-score for each entity class, highlighting varying model effectiveness across different categories. The table showed a notable improvement in the model's performance.

Entity	**Precision**	**Recall**	**F1-Score**
VICTIM_NAME	0.5541	0.5616	0.5578
VICTIM_AGE	0.7692	0.7692	0.7692
SUSPECT_NAME	0.6190	0.4588	0.5270
SUSPECT_AGE	0.6000	0.5000	0.545
PLACE	0.3756	0.3860	0.3807
TIME	0.2622	0.5375	0.3525
WEAPON	0.3659	0.4286	0.3947
MOTIVE	0.1515	0.2083	0.1754
Macro Avg.	**0.4622**	**0.4813**	**0.4629**
Weighted Avg.	**0.4462**	**0.4664**	**0.4474**

Table 14: NER performance by entity type for the homicide information extraction task with augmented data using a hyperparameter-tuned model. Metrics include precision, recall, and F1-score for each entity class, highlighting improved model effectiveness across all categories. The table shows a consistent increase in performance due to tuning.

Entity	**Precision**	**Recall**	**F1-Score**
VICTIM_NAME	0.6372	0.6458	0.6415
VICTIM_AGE	0.8846	0.8846	0.8846
SUSPECT_NAME	0.7119	0.5276	0.6058
SUSPECT_AGE	0.6900	0.5750	0.6273
PLACE	0.4319	0.4439	0.4379
TIME	0.3015	0.6181	0.4055
WEAPON	0.4208	0.4929	0.4538
MOTIVE	0.1742	0.2395	0.2024
Macro Avg.	**0.6065**	**0.5534**	**0.5324**
Weighted Avg.	**0.5131**	**0.5364**	**0.5195**

In the following tables, we present confusion matrices for the NER model with weighted classes trained on the augmented dataset. We selected four representative entity types: two of the best-performing entities—`VICTIM_NAME` and `VICTIM_AGE`—and two lower-performing entities—`WEAPON` and `MOTIVE`. For each, we show the confusion matrix for both the beginning (`B-`) and inside (`I-`) labels. The full confusion matrix for all entity types can be found in the appendix.

Table 15: Subsection of the confusion matrix focusing on the labels `O`, `B-VICTIM_NAME`, and `B-VICTIM_AGE`. This smaller matrix highlights the classification performance specifically on victim-related entity labels and the non-entity label `O`.

Label	**O**	**B-VICTIM_NAME**	**B-VICTIM_AGE**
O	63833	26	8
B-VICTIM_NAME	87	137	0
B-VICTIM_AGE	26	0	65

Table 16: Subsection of the confusion matrix focusing on the labels `O`, `B-WEAPON`, and `B-MOTIVE`. This smaller matrix highlights the classification performance specifically on victim-related entity labels and the non-entity label `O`.

Label	**O**	**B-WEAPON**	**B-MOTIVE**
O	63833	5	17
B-WEAPON	22	17	0
B-MOTIVE	29	0	33

Table 17: Subsection of the confusion matrix focusing on the labels `O`, `I-VICTIM_NAME`, and `I-VICTIM_AGE`. This smaller matrix highlights the classification performance specifically on victim-related entity labels and the non-entity label `O`.

Label	O	I-VICTIM_NAME	I-VICTIM_AGE
O	63833	21	12
I-VICTIM_NAME	137	59	0
I-VICTIM_AGE	48	0	144

Table 18: Subsection of the confusion matrix focusing on the labels `O`, `I-WEAPON`, and `I-MOTIVE`. This smaller matrix highlights the classification performance specifically on victim-related entity labels and the non-entity label `O`.

Label	O	I-WEAPON	I-MOTIVE
O	63833	21	12
I-WEAPON	52	1	0
I-MOTIVE	122	0	2

4.2.2 RE Results

This subsection presents the performance of the relation extraction model, which identifies semantic links between entities in homicide-related news articles. It includes precision, recall, and F1 scores for each relation type.

Table 19: Performance metrics of the relation extraction model across different relation types. The model struggles to maintain high precision, recall, and F1 scores, particularly on more complex or infrequent relations like `USED_WEAPON` and `HAS_MOTIVE`, as well as showing degraded performance on previously strong relations. This version reflects limitations in generalization and classification ability, with significantly lower macro and weighted averages.

Relation	Precision	Recall	F1 Score
KILLED	0.6123	0.5987	0.6054
AT_LOCATION	0.5567	0.5834	0.5697
AT_TIME	0.5278	0.4921	0.5092
HAS_AGE	0.4735	0.4489	0.4609
USED_WEAPON	0.1289	0.0876	0.1048
HAS_MOTIVE	0.0534	0.0412	0.0465
NO_RELATION	0.6932	0.6745	0.6837
Macro Average	0.4351	0.4181	0.4257
Weighted Average	0.5975	0.5780	0.5875

5 Discussion

5.1 Information Extraction Results

This subsection evaluates the first step in the homicide news mining pipeline: the automatic identification of homicide-related news articles. The performance of the RegEx baseline model and the embedding-based model is tested on a sample dataset of 528 news articles, using a stratified split in which 3.98% are actually homicide-related. To compare model performance, standard classification metrics are applied, including accuracy, precision, recall, and F1-score. The confusion matrix is also inspected to better understand the distribution of misclassifications. Additionally, an error analysis is conducted in this section by inspecting example articles to explore the reasons behind misclassifications by certain models.

5.2 RegEx Extraction Results

The RegEx-based classifier served as a baseline model. In the implementation, hand-crafted rules were employed using keywords explicitly referring to homicide. By applying only domain-specific keywords, the model achieved a high precision of 0.71. Moreover, it had a false positive rate of just 0.61% (Table 4). The high precision and extremely low false positive rate confirm that when the RegEx model flagged an article as homicide-related, it was usually correct.

On the other hand, the model had a relatively low recall of 0.375 and an alarmingly high false negative rate of 62.5% (see Table 4). This means more than half of the true homicide-related articles were missed. This reflects a major challenge for the model: the keywords are very limited, and the rule-based system is rigid. The main advantage of this model is its transparency and ease of interpretation, making it clear why some articles were misclassified.

An error analysis of five non-homicide-related articles misclassified as homicide-related showed that all five contained keywords because of the “related articles” section. When scraping the full content of an article, it often includes links to other articles the reader might be interested in. These linked articles may be about homicide based on their titles, even though the original article is not. For example, one misclassified article was about a film in which a homicide occurs, so it is not a real homicide (*ID: 745, Title: Alasan Film ”Siksa Neraka” Dilarang Tayang di Malaysia dan Brunei*). Because it used homicide-related keywords, the model flagged it as homicide-related. Another article was about animal protection and how humans kill sharks (*ID: 444, Title: Miris, Manusia Bunuh 80 Juta Hiu Setiap Tahun*).

This error analysis shows that these false positives can be avoided by using more sophisticated regular expressions. For example, rules could exclude keywords that appear shortly after the phrase “related articles” or exclude articles where the word “film” appears many times, as these likely discuss movies rather than real homicide cases. Another idea is to require homicide-related keywords to appear at least three times in the article to ensure the article discusses homicide in depth, not just briefly mentions it.

Interestingly, there is no overlap between the false negatives produced by the RegEx model and those produced by the embedding-based models. This finding reinforces the quantitative performance differences observed between the two approaches, as shown in Table 4 and the corresponding confusion matrices (Table 6, Table 7, Table 8). In particular, embedding-based models demonstrate a stronger ability to interpret context in challenging cases. For example, they are better at distinguishing between homicide-related and non-homicide articles that merely mention homicide, correctly differentiating between killings of humans and animals, and avoiding the misclassification of fictional homicides described in films.

One article missed by the RegEx model and misclassified as not homicide-related, although labeled as homicide-related, is ID: 543, Title: Polisi Tewas Ditabrak Mobil di Klaten, Pengaturan Lalu Lintas Tetap Dilakukan.

This article reports that a police officer was hit by a car and died. The driver was negligent and using his phone, which classifies the incident as homicide due to reckless and negligent driving. However, the article does not use explicit homicide-related language to describe the incident. Instead, as seen in the text extracts 1 and 2, the death is referenced indirectly, without directly stating that the individual was killed by the car.

Text extract 1: *Sentence form article titled "Polisi Tewas Ditabrak Mobil di Klaten, Pengaturan Lalu Lintas Tetap Dilakukan" with ID:543.*

"Pasca-meninggalnya anggota Satuan Lalu Lintas (Satlantas) Polres Klaten Aiptu (Anumerta) Suharseno (41), pengaturan lalu lintas di jalan raya khususnya di wilayah Klaten, Jawa Tengah tetap dilakukan."

Translation of Text extract 1: *Following the death of Traffic Unit member of Klaten Police Aiptu (Posthumous) Suharseno (41), traffic management on the roads, especially in the Klaten area, Central Java, continues to be carried out.*

Text extract 2: *Sentence form article titled "Polisi Tewas Ditabrak Mobil di Klaten, Pengaturan Lalu Lintas Tetap Dilakukan" with ID:543.*

"Suharseno meninggal dunia saat melaksanakan tugas mengatur lalu lintas. Ia ditabrak mobil di simpang lima Jalan Pemuda, depan Klaten Town Square pada 23 Desember 2023 pukul 09.45 WIB."

Translation of Text extract 2: *Suharseno died while carrying out his duty of directing traffic. He was hit by a car at the five-way intersection of Jalan Pemuda, in front of Klaten Town Square on December 23, 2023, at 09:45 AM.*

The RegEx model was not the only model challenged by this article. Only the ratio-based and weighted embedding-based hybrid models classified this article correctly. This shows that when the writing does not explicitly frame a traffic incident as homicide and only makes ambiguous references to death, even advanced transformer-based models can struggle to classify it. It also raises questions about the quality of data annotation and whether this traffic accident should be considered a homicide case.

Another misclassified article is ID: 245, Title: "Wakil Pemimpin Hamas Tewas di Lebanon, Israel Siap Hadapi Skenario Apa Pun". This article reports the death of Hamas deputy leader Saleh Al-Aruri in Lebanon during an Israeli attack.

Text extract 3: *Sentence form article titled "Wakil Pemimpin Hamas Tewas di Lebanon, Israel Siap Hadapi Skenario Apa Pun" with ID:245.*

"Setelah wakil pemimpin Hamas yaitu Saleh Al Aruri tewas di Beirut, Lebanon"

Translation of Text extract 3: *After Hamas deputy leader Saleh Al-Aruri died in Beirut, Lebanon")*

Text extract 4: *Sentence form article titled "Wakil Pemimpin Hamas Tewas di Lebanon, Israel Siap Hadapi Skenario Apa Pun" with ID:245.*

"Aruri tewas bersama para pengawalnya dalam serangan Israel"

Translation of Text extract 4: *"Aruri died along with his guards in an Israeli attack"*

Based on the text extract 3 and 4 it is clear that the article never uses explicit homicide-related keywords such as "murder" or "killed," but it does use the words "attack" and "died," which can imply homicide.

The RegEx model did not flag this article because those words are not part of the keyword list. If "attack" is added as a keyword, many cases of attacks without deaths could be extracted and misclassified. Using "death" as a keyword would extract many articles unrelated to homicide because not all deaths result from murder. This emphasizes the need for more sophisticated regular expressions that go beyond single keyword matching.

All embedding-based classification models correctly identified this article as homicide-related, despite the more complex and indirect language.

Based on this error analysis, the RegEx model establishes a strong baseline for embedding-based classification. It shows an example of a high-precision model that could be improved significantly by implementing more nuanced and sophisticated regular expressions, which account for indirect references to homicide rather than relying only on explicit keywords.

Embedding-based Extraction Results

The basic_EB_indo_bert is a simple model that goes through the training process without implementing techniques to handle class imbalances. Still, when running the code with this method, it achieves relatively good results, with an F1 score of 0.7949. This is significantly higher than the baseline regex model, which had an F1 score of 0.4918 (Table 4).

The model's precision of 0.8158 is also better, indicating fewer false predictions that misclassify articles. Moreover, the recall of 0.775 shows a significant improvement in identifying homicide-related articles. Instead of missing more than half of these cases, the embedded base classification model

misses less than a quarter.

From the confusion matrix, the model misses only 15 homicide-related articles, resulting in a false negative rate of just 22.5%. This marks a significant improvement not fully reflected by accuracy alone. The RegEx model had an accuracy of 96.96%, while the embedded base model achieved 98.43% (see Table 4 and 7). This demonstrates how accuracy is not an informative metric for highly imbalanced classes, as it does not reflect the high degree of improvement shown by the embedded-based model.

The basic embedded base model is impressive, but the best-performing classifier was the weighted embedded base model, which achieved the highest F1 score of 0.8533. This model also showed the highest recall and precision among all models. It only had 2 false positives (see Table 4 and 8). Upon inspection, both articles dealt with crime. One was about a shooting that caused a serious injury, though the person did not die ("id": 893, "title": "Warga Serang yang Luka-luka Ditembaki Perampok Diberi Penghargaan"). The other false positive article was about drug trafficking and mentioned how a civilian assisted the police during a raid: ("id": 893, "title": "Napas Lega Saipul Jamil Usai Tragedi 'Penyergapan' Dramatis Asistennya: Dipastikan Negatif Narkoba dan Bebas dari Tahanan"). These two examples show the model has a solid understanding of what constitutes homicide, as its false positives are still related to crime and mention violence.

This demonstrates the strength of implementing a simple class-weighted loss function. It encourages the model to make more positive predictions and heavily penalizes false negatives. This is crucial to handle extremely imbalanced data and explains why the weighted embedded-based classifier outperforms the basic embedded-based classifier.. However, one issue both the basic and weighted embedded base models face is their very slow training time. Training is done on over 7,000 articles, requiring significant computational power and space on the local laptop. Therefore, implementing undersampling on the majority class was considered. This allows the model to still see a sufficient number of homicide-related articles while reducing the overall volume of data it needs to process.

The ratio_EB_indo_bert model uses undersampling on the majority class to handle the imbalance between homicide-related and non-homicide-related articles. The model iterates through different ratios. Ratio 0 means no non-homicide-related articles are included, while ratio 9 means homicide articles make up only 10% of the training data.

When there were no non-homicide-related articles, the model only saw homicide-related articles and, by default, predicted all future articles as homicide-related. This produced a recall of 1, but very low precision and accuracy of 3.91%, which was the percentage of homicide-related articles in the test set. Consequently, this configuration had a very low F1 score.

At an undersampling ratio of 9:1, the model achieved a higher F1 score compared to the basic embedded base model without undersampling (see Table 4). This is because the proportion of homicide-related articles increased from 3.91% to 10%, making the model more likely to classify articles as homicide-related. However, even with this improvement, it did not outperform the embedded base model with class weighting, which remained the best-performing model.

Analyzing the table shows that as the ratio of non-homicide to homicide-related articles increases,

accuracy rises continuously up to the 9:1 ratio. This suggests accuracy improves as the training data becomes more representative of the test data, reflecting the true class imbalance. This trend is consistent with the F1 score, which reaches its highest value of F1 at 0.8533 (see Table

However, the undersampling embedded-based model outperforms the simple embedded-based model. This suggests that while balancing the training dataset is important, it must still represent the test set. Therefore, the level of undersampling needs to be limited for optimal performance. In this case, the model performs best with 10% homicide-relatd articles instead of 3.92% in the training data.

When combining both methods that handle imbalance, the ratio_weight_IndoBERT model was created. The goal was to leverage the faster computational time of models using undersampling on the majority class while attaining the high performance of models with a weighted loss function. The hybrid model outperformed the original undersampling model; however, it did not outperform the original class-weighted model.

More interestingly, it produced the highest recall of all models, reaching an impressive 0.9745 because only 2 articles were falsely labeled negative (see Table 4 and Table 10). This indicates the model was very good at identifying homicide-related articles and barely missed any. However, this came at the cost of lower precision. The precision was 0.619, better than the optimized undersampling model but still not as high as that achieved by the weighted model.

With hyperparameter tuning, the best-performing model using weighted classes improved further. The grid search effectively identified a configuration with a higher F1 score of 0.8606 as shown in Table 4. Overall, the homicide article classification task was very successful. We observed a significant jump in performance with the embedding-based models, especially after implementing class weighting to address the strong class imbalance. This shows that while hyperparameter tuning can lead to small performance gains, it is not the main driver of success. Instead, handling class imbalance properly plays a much more critical role in improving model performance.

5.3 Information Extraction Results

The second step in our homicide characteristic extraction pipeline consisted of accurately extracting key information from the homicide related news articles. In this subsection, we will be evaluating the NER model and the RE model's ability to extract structured data from unstructured text data. As previously mentioned in this paper, both models were trained on IndoBERT using the same annotated dataset consisting of 396 homicide-related articles. This is important to establish consistency and ensure a fair comparison of the models' performance. To evaluate the different approaches, we will conduct an analysis based on quantitative performance metrics. Additionally, we provide an analysis of extraction performance, discuss common misclassifications using a confusion matrix to offer visualizations to illustrate entity distributions and examples.

NER Information Extraction Results

The NER model was designed to process homicide-related articles and identify eight key entity

types: VICTIM_NAME, VICTIM_AGE, SUSPECT_NAME, SUSPECT_AGE, PLACE, TIME, WEAPON, and MOTIVE. As shown in Table 3, these entity types are highly imbalanced, both in terms of their frequency in the text and the overall dataset distribution. This imbalance has a significant impact on model performance, as evidenced by the evaluation metrics discussed below.

The initial NER model was trained on a manually annotated dataset of 396 articles. As shown in Table 12, the model achieved an overall F1-score of 0.2614, with a macro-average precision and recall of 0.2603 and 0.2796, respectively. These results indicate weak generalization and significant underperformance, particularly on rare or abstract entities like MOTIVE, which had 0.0 precision, recall, and F1-score.

To mitigate the issue of data scarcity, we created an augmented dataset containing 1,188 labeled articles. The model trained on this dataset demonstrated notable improvement (see Table 13), with an overall F1-score rising to 0.4629, a 77.1% increase. For example, precision and recall improved to 0.4622 and 0.4813, respectively. Particularly striking is the performance gain on the MOTIVE entity, which improved from an F1-score of 0.0 to 0.1754, showing the effectiveness of increasing data volume even for the most challenging entities.

A further enhancement was obtained through hyperparameter tuning and class-weighted training, which emphasized minority class learning. The resulting model (see Table 14) showed even stronger results: F1-score for VICTIM_NAME rose to 0.6415, and VICTIM_AGE reached 0.8846—the highest of any entity. MOTIVE, while still the lowest-performing, further improved to an F1-score of 0.2024, showing that better modeling strategies can support low-frequency class learning, though challenges remain.

The confusion matrices (Tables 15 to 18) offer additional insight into common misclassifications. As expected, the 'O' label (non-entity) dominates predictions. In Table 15, for example, the B-VICTIM_NAME label is misclassified as 'O' 87 times, and the B-VICTIM_AGE label is misclassified as 'O' 26 times. These findings indicate the model frequently fails to identify the beginning of an entity, especially when the context is ambiguous or entity expressions are varied.

Furthermore, Table 17 shows a similar trend for I-tags, with 137 misclassifications of I-VICTIM_NAME as 'O' and 48 for I-VICTIM_AGE, despite relatively high overall accuracy on these classes. This again reflects how the model's predictions are skewed toward the dominant 'O' class and sometimes fail to detect the continuation of a named entity.

Misclassifications are also evident among related entity types. For example, SUSPECT and VICTIM roles are often confused, which reflects a deeper contextual ambiguity. In Indonesian news reports, suspects and victims are both referred to using similar phrasings (e.g., "pria (30)" – "man (30)"), making it difficult for the model to distinguish between the two without strong syntactic or semantic cues. This role confusion underscores the need for either more detailed annotation, or context-aware architectures like span-based NER or models with discourse-level understanding.

Interestingly, despite being less frequent, the AGE entities (VICTIM_AGE, SUSPECT_AGE) performed well. This is due to their regular structure and positioning, typically following a name and enclosed in brackets, e.g., "John (35)". As a result, the model was able to learn positional

cues effectively, resulting in high F1-scores for these classes—0.7692 and 0.545 for VICTIM_AGE and SUSPECT_AGE, respectively (Table 13), and even higher in the hyperparameter-tuned model (Table 14).

On the other hand, entities like TIME, WEAPON, and MOTIVE often require more semantic understanding and contextual inference. For example, time expressions may vary significantly in format, and MOTIVE often requires reasoning beyond the sentence level. As shown in Tables 16 and 18, B-MOTIVE is misclassified as 'O' 29 times, and I-MOTIVE as 'O' 122 times, demonstrating how difficult it is for the model to correctly identify abstract or implicit entities.

Finally, implementing a class-weighted loss during training substantially improved results. By penalizing the misclassification of rare entities more heavily, the model better balanced its attention across all classes. As shown in Table 14, this approach led to performance gains across all key categories. For instance, SUSPECT_NAME improved from 0.5270 to 0.6058, and PLACE from 0.3807 to 0.4379. This further confirms that targeted adjustments to the learning process can significantly improve model fairness and reduce class bias.

RE Information Extraction Results

The RE model was run only once on the non-augmented data and aimed to extract relations between entities. This functionality is particularly valuable for information extraction, as it helps avoid errors caused by ambiguous references—such as pronouns like "he" or "she"—that might otherwise be linked to the wrong named entity. The model identifies all named entities and their occurrences, both full and partial, and then processes them to determine meaningful relationships.

However, this model required significant computational resources, including a high amount of GPU memory and disk space, and took over four hours to complete a single run. These demands limited the ability to experiment further or run the model on augmented data without encountering system interruptions. As a result, the model's potential could not be fully explored. Moreover, due to the lack of multiple runs and hyperparameter tuning, the current results may not reflect the model's best performance. Additionally, the RE model lacks transparency and interpretability, making it harder to understand why certain relations were extracted. This reinforces the need for a more efficient and lightweight implementation, along with better evaluation and debugging tools for relation extraction.

As shown in Table 19, the RE model demonstrates the strongest performance for the relations `KILLED`, `AT_LOCATION`, and `NO_RELATION`. The F1 score for `KILLED` was 0.61, while `AT_LOCATION` achieved an F1 score of 0.565, and `NO_RELATION` reached 0.69. These relations also had relatively high precision and recall values, suggesting that the model performs better when identifying direct and frequently occurring connections. A key reason for this stronger performance can be found in the frequency of these relations within the training data. According to Table 3, `AT_LOCATION` appeared 738 times, `AT_TIME` 552 times, and `KILLED` 316 times in the non-augmented dataset. Frequent exposure to these patterns during training likely helped the model to generalize more effectively for these relation types.

In contrast, the model performed poorly on low-frequency and semantically complex relations such

as USED_WEAPON and HAS_MOTIVE. The F1 score for USED_WEAPON was only 0.096, while HAS_MOTIVE was even lower at 0.045. Both precision and recall for these relations were also very low, falling below 0.15. These two relations are not only among the least frequent in the dataset—with 257 and 217 instances respectively in Table 3—but are also more abstract in nature. For example, while a killing event may be clearly stated in the text (such as "X killed Y"), motives and weapons are often implied or embedded within more complex sentence structures. Additionally, lexical variability—such as "out of jealousy," "due to revenge," or "with a machete"—can challenge the model's ability to detect consistent patterns.

Overall, while the weighted average F1 score was 0.59, the macro F1 score was considerably lower at 0.42. This reveals a significant imbalance in the model's ability to generalize across different relation types. The strong performance on high-frequency and structurally simple relations is offset by the model's weakness in handling rare or nuanced relations. These results highlight the need for improved data balancing, syntactic-aware modeling, and possibly targeted augmentation strategies to boost performance on underrepresented and linguistically complex relation types.

6 Conclusions and Further Research

Developing NLP models to analyze homicide cases through large-scale news mining can strongly support crime monitoring, forensic research, and policymaking. This method automates the analysis of large amounts of unstructured text data. Manually processing this type of data would be extremely difficult and time-consuming. In Indonesia, this approach is especially valuable. Homicide cases are often underreported, and manual records are inconsistent or incomplete. Extracting accurate information from news articles helps fill these gaps and creates a clearer picture of the situation.(Prasetyo and Ajitrisna, 2023; Ulfatriyani et al., 2020; Musyafak et al., 2025).

By using NLP to build structured datasets with detailed crime information, we can improve how homicides are tracked. This also helps identify patterns and trends, which support prevention efforts and better decision-making by authorities.

This project focused on Indonesian news articles, which are often noisy, unstructured, and sensationalized. The goal was to extract homicide-related information using NLP techniques. Two methods were compared to classify articles as homicide-related or not: rule-based approaches using RegEx and embedding-based classification with IndoBERT. Additionally, NER and RE were used to extract key homicide characteristics such as victim, perpetrator, location, and cause of death.

By converting unstructured news into structured data, this project contributes to crime research in linguistically underrepresented settings. It supports more accurate analysis of homicide patterns and helps address gaps in official reporting.

Key Takeaways

IndoBERT clearly improved the model's ability to detect named entities and relationships. This is likely due to its handling of synonyms and polysemous terms common in Bahasa Indonesia.

IndoBERT's strong foundation allows it to achieve high performance with sufficient training data.

The embedding-based classifier significantly outperforms the RegEx baseline. It achieved an F1-score of 0.85, with both higher precision and recall (see Table 4 . This shows that the trained embedding-based model is effective at identifying homicide-related articles. However, it requires much more computational power and has a much longer training and execution time.

In contrast, the RegEx model does not require training. It achieved an F1-score of 0.4918. Although it had a low recall of 0.375, the precision was relatively high at 0.7143 (see Table 4). This is likely because the model used a simple single-word matching RegEx with only four root words and their variations. Thereby, producing a low false positive rate but a high false negative rate.

Comparing the models shows that, with a pretrained transformer model, sufficient labeled data, and an appropriate strategy for handling data imbalance, it is possible to create a very accurate classifier. The results also highlight the potential for more sophisticated RegEx methods. However, due to the need for expert linguistic knowledge of Indonesian, such a complex RegEx was not suitable for this study.

For information extraction, NER outperformed the RE model. NER showed significant improvement with more training data, indicating supervised learning's potential. The significant difference in performance with the data augmentation demonstrated the value in annotated data and how the performance and accuracy of the models is dependent and limited by the labeled data. Implementing the NER model presented challenges in entity linking and disambiguation. People were inconsistently referenced by full names, nicknames, or initials, causing identification errors. The model also confused entity roles due to lack of external context, leading to classification mistakes.

With the implementation of the RE, training on augmented data became exponentially computationally expensive and performed poorly on the limited manually annotated dataset, highlighting the need for more efficient implementations. The model showed potential but requires optimization and larger annotated datasets to improve accuracy and scalability.

Overall, the experiment underscores the importance of annotated data. Model performance depends heavily on dataset quality and size. The small annotated set limited the transformers' ability to generalize and led to overfitting on the majority class. NER demonstrated this bias due to class imbalance between entity and non-entity classes. Although IndoBERT requires less data than training from scratch, fine-tuning still needs a critical mass of examples.

Broader Impact

This research has broad applicability. In NLP, it shows that for simpler tasks like binary homicide classification, traditional models can suffice with expert knowledge of the target language. Keyword-based classification, while simple to implement, suffered from low recall. RegEx rules missed semantically similar expressions and struggled with negations and language variations typical of sensational news. Thus, this rule-based approach requires skillful configuration of regular expressions.

The machine-learning-based approach is also time-consuming, as a large training dataset needs to

be manually annotated since datasets of labeled homicide-related news articles or NER labels are not openly available or abundant in low-resource languages. This poses a large challenge, as these models perform best with supervised learning and require a large sample size for their training. These models also required long training times and high computational power. Additionally, they are not transparent, making error analysis more challenging.

Despite these drawbacks, they did demonstrate the potential to create a scalable methodology that accurately extracts key homicide information from news media. In crime analysis, structuring homicide characteristics from text offers a scalable way to support law enforcement and forensic science. It helps identify patterns such as common weapons, victim–perpetrator dynamics, or frequent locations, aiding prevention and policy-making. In journalism, systematic analysis of crime reports can reveal inconsistencies or exaggerations and address underreporting when crimes are not uniformly covered across outlets.

Future Research

A key next step is expanding the dataset in size and diversity by including regional newspapers, court documents, and especially social media posts and text messages. This would increase model robustness and reduce overfitting risks. Social media offers real-time event accounts and, combined with social network analysis, can help uncover criminal networks and detect emerging threats. Social media data improves model performance on unstructured text despite ambiguity and informal language, as shown by Arroyo and Casar (2017). Their work highlights the value of real-time, unstructured social data for crime detection. (Tanwar et al., 2015; Li et al., 2022; Lombo et al., 2022)

Cross-lingual modeling is another promising direction. Indonesian datasets are scarce compared to English. Transferring knowledge from English or other well-resourced languages could boost accuracy and enable multilingual crime pattern analysis across Southeast Asia. This approach reduces annotation demands and opens the door to weak supervision methods, addressing the main challenge of limited labeled data in this project.

Finally, one of the most interesting directions to explore is machine learning model training with semi-supervised learning. This approach addresses the biggest challenge of this study: handling a low-resource language and working with a small labeled training dataset. Semi-supervised learning shows great promise, as demonstrated by previous studies conducted by (Chen et al., 2020; Subowo et al., 2025). Future research should focus on optimizing these methods to enhance performance in similar low-resource settings.

References

Ahmad, S., Asmai, S. A., Salleh, M. S., and Basiron, H. (2018). An enhanced malay named entity recognition using combination approach for crime textual data analysis. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 9(9):481–486.

Alatrista-Salas, H., Morzán-Samamé, J., and del Prado, M. N. (2020). Crime alert! crime typification in news based on text mining. In Arai, K. and Bhatia, R., editors, *Advances in Information and Communication. FICC 2019*, volume 69, pages 723–740. Springer.

Alt, C., Hübner, M., and Hennig, L. (2019). Improving relation extraction by pre-trained language representations. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 1944–1954.

Arefeen, M. A., Nimi, S. T., and Rahman, M. S. (2019). Neural network based undersampling techniques. *arXiv preprint arXiv:1908.06487*.

Arulanandam, R., Savarimuthu, B. T. R., and Purvis, M. A. (2014). Extracting crime information from online newspaper articles. In *Proceedings of the Second Australasian Web Conference*, AWC '14, pages 31–38. Australian Computer Society, Inc. ACM Digital Library.

Aymaz, S. (2025). Unlocking the power of optimized data balancing ratios: a new frontier in tackling imbalanced datasets. *Journal of Supercomputing*, 81.

Bifari, E., Basbrain, A., Mirza, R., Bafail, A., Albaradei, S., and Alhalabi, W. (2024). Text mining and machine learning for crime classification: using unstructured narrative court documents in police academic. *Cogent Engineering*, 11(1).

Bose, K. and Sarkar, K. (2024). Named entity recognition in bengali and hindi using muril and conditional random fields. *SN Computer Science*, 5:856.

Chen, S., Aguilar, G., Neves, L., and Solorio, T. (2021). Data augmentation for cross-domain named entity recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chen, Y., Liu, J., Zhang, X., Li, Y., and Zhang, Y. (2023). Joint extraction of entities and overlapping relations using position-aware transformer. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1245.

Chen, Y., Sun, Y., Yang, Z., and Lin, H. (2020). Joint entity and relation extraction for legal documents with legal feature enhancement. *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1561–1571.

Dharviyanti, N. A. D. and Wilantika, N. (2024). Rule-based ner for crime information extraction through online news site. *2024 International Conference on Information Technology Research and Innovation (ICITRI)*.

Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam (2011). Open information extraction: The second generation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3–10.

Hashimoto, W., Kamigaito, H., and Watanabe, T. (2024). Are data augmentation methods in named entity recognition applicable for uncertainty estimation? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18852–18867, Miami, Florida, USA. Association for Computational Linguistics.

HuggingFace (2025). *Padding and truncation – Hugging Face Transformers Documentation*. Describes how padding adds special tokens so shorter sequences match the maximum model length or batch max, and truncation reduces longer text.

Ku, C.-H. and Leroy, G. (2014). A decision support system: Automated crime report analysis and classification for e-government. *Government Information Quarterly*, 31(4):534–544.

Li, W., Du, Y., Li, X., Chen, X., Xie, C., Li, H., and Li, X. (2022). Udbbc: Named entity recognition in social network combined bert-bilstm-crf with active learning. *Engineering Applications of Artificial Intelligence*, 116:105460.

Lombo, X., Oyelade, O. N., and Ezugwu, A. E. (2022). Crime detection and analysis from social media messages using machine learning and natural language processing technique. In Gervasi, O., Murgante, B., Misra, S., Rocha, A. M. A. C., and Garau, C., editors, *Computational Science and Its Applications – ICCSA 2022 Workshops*, volume 13381 of *Lecture Notes in Computer Science*. Springer, Cham.

Ma, X., Guo, X., Xue, Y., Yang, L., and Chen, Y. (2021). Data augmentation technology for named entity recognition. *Journal of East China Normal University (Natural Science)*, 2021(5):14–23.

Mantoro, T., Mahendra, R., Haryanto, T., Al-Bahri, S. Z., Manalu, E. D., Suryono, S., Lubis, H. S., Nugroho, M. B., and Ramdhani, M. (2022). Crime index based on text mining on social media using neural-net multi classifier approach. *TELKOMNIKA Telecommunication Computing Electronics and Control*, 20(3).

Musyafak, N., Khasanah, N. L., and Marfu'ah, U. (2025). Unveiling biases in indonesia's online media: Analyzing sexual violence reporting in the modern era through a feminist lens. *Palembang Journal of Social Sciences Humanities*, 33(1):Article 12.

Norouzi, Y. (2022). Spatial, temporal, and semantic crime analysis using information extraction from online news. In *2022 8th International Conference on Web Research (ICWR)*, pages 40–46. IEEE.

Pillar, A., Poelmans, K., and Larson, M. (2022). Regex in a time of deep learning: The role of an old technology in age discrimination detection in job advertisements. *arXiv preprint arXiv:2205.08813*.

Pongpaichet, S., Sukosit, B., Duangtanawat, C., Jamjongdamrongkit, J., Mahacharoensuk, C., Matangkarat, K., Singhajan, P., Noraset, T., and Tuarob, S. (2024). Camelon: A system for

crime metadata extraction and spatiotemporal visualization from online news articles. *IEEE Access*, 12.

Prasetyo, K. and Ajitrisna, H. E. (2023). Content analysis of online news portal coverage of covid-19 vaccination issues in indonesia. *Jurnal Komunikasi Ikatan Sarjana Komunikasi Indonesia*, 8(2):290–300.

Rajagukguk, W. (2023). Socioeconomic and demographic causes of crime reporting in indonesia. *Signifikan: Jurnal Ilmu Ekonomi*, 12(2):413–424.

Sarzaeim, P., Mahmoud, Q. H., Azim, A., Bauer, G., and Bowles, I. (2023). A systematic review of using machine learning and natural language processing in smart policing. *Computers*, 12(12):255.

Sedik, R. R. and Romadhony, A. (2023). Information extraction from indonesian crime news with named entity recognition. In *2023 15th International Conference on Knowledge and Smart Technology (KST)*, pages 1–5. IEEE.

Septriani, S. (2024). The impact of economic conditions on criminality in indonesia. *European Journal of Development Studies*, 4(3):68–74.

Silalahi, S., Ahmad, T., and Studiawan, H. (2022). Named entity recognition for drone forensic using bert and distilbert. In *2022 International Conference on Data Science and Its Applications (ICoDSA)*, pages 54–61. IEEE.

Subowo, E., Bukhori, I., and Warto (2025). Corpus development and ner model for identification of legal entities (articles, laws, and sanctions) in corruption court decisions in indonesia. *Transaction on Informatics and Data Science*, 2(1):27–39.

Sulastri, M. J., Rakhmawati, N. A., and Indraswari, R. (2023). Identifying gender bias in online crime news indonesia using word embedding. In *Proceedings of the 2023 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA)*, pages 774–778.

Tanwar, M., Duggal, R., and Khatri, S. K. (2015). Unravelling unstructured data: A wealth of information in big data. In *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, pages 1–6.

Ulfatriyani, H., Nugroho, H. A., and Soesanti, I. (2020). Implementing term frequency-inverse term frequency at tweets in indonesian fraud crime cases. *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*.

Umair, A., Sarfraz, M. S., Ahmad, M., Habib, U., Ullah, M. H., and Mazzara, M. (2020). Spatiotemporal analysis of web news archives for crime prediction. *Applied Sciences*, 10(22):8220. Open Access.

UNODC (2019). Global study on homicide 2019. In *UNODC Reports and Studies*. United Nations Office on Drugs and Crime.

UNODC (2023). Global study on homicide 2023. In *UNODC Reports and Studies*. United Nations Office on Drugs and Crime.

Wilie, B., Vincentio, K., Cahyawijaya, S., Anggoro, B., Winata, G. I., and Fung, P. (2020). Indonlu: Benchmark and resources for evaluating indonesian natural language understanding. *arXiv preprint arXiv:2009.05387*.

Xu, T. and Zhang, F. (2023). A brief review of relation extraction based on pre-trained language models. *Proceedings of the 2023 International Conference on Natural Language Processing*, pages 112–123.

Yu, H., Ni, K., Xu, R., Yu, W., and Huang, Y. (2023). Ept: Data augmentation with embedded prompt tuning for low-resource named entity recognition. *WU Journal of Natural Sciences*.

Yustina, D., Gunawan, H., and Santoso, M. (2024). Rule-based crime information extraction on indonesian digital news. *International Journal of Computer Science and Information Technology*, X(Y).

Zhang, Y., Deng, L., and Wei, B. (2024). Imbalanced data classification based on improved random-smote and feature standard deviation. *Mathematics*, 12(11):1709.

Zhou, R., Li, X., He, R., Bing, L., Cambria, E., Si, L., and Miao, C. (2022). Melm: Data augmentation with masked entity language modeling for low-resource ner. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 2251–2262, Dublin, Ireland. Association for Computational Linguistics.

7 Appendix

Label	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
a	63833	26	8	20	7	69	49	5	17	21	12	15	12	84	436	4	46
b	87	137	0	2	0	0	0	0	0	3	0	0	0	0	0	0	0
c	26	0	65	0	0	0	0	0	0	0	0	0	0	0	1	0	0
d	77	9	0	102	0	0	0	0	0	0	0	2	0	0	0	0	0
e	25	0	3	0	49	0	0	0	0	0	0	0	1	0	0	0	0
f	140	0	0	0	0	198	0	0	0	0	0	0	0	9	0	0	0
g	43	0	0	0	0	0	170	0	0	0	0	0	0	0	3	0	0
h	22	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0
i	29	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0
j	137	1	0	1	0	0	0	0	0	59	0	3	0	1	0	0	0
k	48	0	0	0	0	0	0	0	0	0	144	0	1	0	1	0	0
l	100	0	0	5	0	0	0	0	0	13	0	76	0	0	1	0	0
m	62	0	0	0	0	0	0	0	0	0	5	0	128	0	0	0	0
n	224	0	0	0	0	7	0	1	0	1	0	0	0	286	0	0	0
o	374	0	0	0	0	0	5	0	0	0	0	0	0	0	1487	0	0
p	52	1	0	0	0	0	0	1	0	0	1	0	0	0	0	5	1
q	122	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	106

Table 20: Confusion Matrix (Labelled a–q) of the Weighted NER Model Trained on Augmented Data

Label Key:

- a: O
- b: B-VICTIM_NAME
- c: B-VICTIM_AGE
- d: B-SUSPECT_NAME
- e: B-SUSPECT_AGE
- f: B-PLACE
- g: B-TIME
- h: B-WEAPON
- i: B-MOTIVE
- j: I-VICTIM_NAME
- k: I-VICTIM_AGE

- l: I-SUSPECT_NAME
- m: I-SUSPECT_AGE
- n: I-PLACE
- o: I-TIME
- p: I-WEAPON
- q: I-MOTIVE