국제학 석사 학위논문

Alternative Hypothesis Retrieval Model for Crime Investigation Analysis Using Argument Mining

논증 마이닝 기법을 활용한 범죄 수사 경합가설 탐색에 관한 연구

박 성 미 (Park, Sung Mi)

국제학과(Department of International Studies)

정보법과학전공(Major in Legal Informatics and Forensic Science)

한림대학교 대학원

(Graduate School, Hallym University)

국제학석사학위논문

Alternative Hypothesis Retrieval Model for Crime Investigation Analysis Using Argument Mining

2020

박성미

국제학 석사 학위논문

Alternative Hypothesis Retrieval Model for Crime Investigation Analysis Using Argument Mining

논증 마이닝 기법을 활용한 범죄 수사 경합가설 탐색에 관한 연구

박 성 미 (Park, Sung Mi)

국제학과(Department of International Studies)

정보법과학전공(Major in Legal Informatics and Forensic Science)

한림대학교 대학원

(Graduate School, Hallym University)

장 윤 식, 노 기 영 교수지도

국 제 학 석사 학위논문

박 성 미 의 석사 학위논문을 합격으로 판정함

2020년 12월 21일

심사위원장 안 정 민

_____

심사위원 장 윤 식

_____

심사위원 김 기 범

_____

심사위원 노 기 영

Table of Contents

List of Tables

List of Figures

I. Introduction

1. Background

In recent years, artificial intelligence and data analysis technologies in legal domains have steadily increased. From efficient case citation software to fully-functional AI lawyers [1], legal technology has grown to encompass various solutions that not only help human legal experts but augment the scope of their ability. This phenomenon can be observed no only in legal disputes but also within the context of police investigations.

In 2017, the South Korean Police announced the development of CLUE (Crime Layout Understanding Engine), a crime analysis, and detection system using Big Data and AI technology [2]. CLUE uses crime patterns extracted from official crime documents to find similar cases and predict potential suspect list or their location.

The importance of developing a system that supports investigators has been emphasized with the recent amendments to the Korean Criminal Procedure Act (Implemented in July 2020). The new act support court-oriented trials, limiting the admissibility of the prosecution's suspect interrogation reports as evidence (Article 312 of the Amended South Korean Procedure Act)[1]. The amendment also reorganizes the investigation structure so the police can conduct investigations

[1] Before the amendment, statements of the suspect to the prosecution were admissable as evidence even when the suspect retracted the statement afterwards. This served as a powerful tool to form a judge's opinion favorable to the prosecution. The new amendment considers the interrogation report only admissable if the suspect does not deny and confirms the content.

for all crimes independent of prosecution (Article 197)[2] and gives the police the authority to close cases without sending the case to the prosecutor (Article 245). Through these changes, the importance of thorough evidence gathering and verification at the investigation stage for the police has become more crucial than ever.

The investigation process can be understood as the merge of two major loops: the foraging process, which effectively collects and analyzes evidence, and the sense-making process [3]. The sense-making process is a cycle of generating hypotheses to reconstruct crime events and evaluate the hypotheses by searching for support, relations between hypothesis and evidence (See II.1 1) for details).

However, current data analysis technology or Digital Forensics tools mostly focus on acquiring, analyzing, and verifying information from their sources, leaving the sense-making process solely to the human investigator. The lack of technical support can pose a problem, especially with the enactment of the new criminal procedure legislation: the police investigators will be under higher scrutiny while suffering through lack of human resources[3].

Thus, an automatic sense-making support system for investigators is required to maintain legal security and uphold justice.

---

[2] Until this amendment, the prosecution were given full authority for both investigation and prosecution, causing a monopoly of investigative and prosecutorial power. Prosecutors in high-profile cases were often pressured of persuaded to follow the lead of prosecutors in higher office, who were influenced by the rulling party or high-ranking government officials. This political arrangement lead to inevitable corruption and violation of justice [87].

[33] In order to respond to the changes of the Criminal Procedure Act, the police newly established investigative examiners (수사심사관) in charge of case analysis and case supervision, and supplemented the warrant examiner (영장전담심사관) system. However, due to the lack of manpower and extensive data that need to be reviewed and analyzed, the efficiency and practicality are still in question [88].

2. Purpose of Research

This research aims to provide a tool that can accelerate and enhance the investigator's sense-making process. As mentioned in the previous section, the sense-making process consists of generating hypotheses from gathered evidence and evaluating the hypotheses. Hypotheses can be understood as conclusions inferred from a collection of information that is supported by arguments [3]. Sense-making systems used in practice are usually supporting tools that help the users structure their own logic [4]. Most of them are argument visualization tools that are not linked to a knowledge base and do not provide automated analysis [4], [5]. Some tools enable the users to build arguments of a case using underlying argumentation logic [6], [7]. Other tools implement probabilistic methods such as Bayesian Networks to evaluate the evidence and arguments based on the user's probability or degree of belief [8].

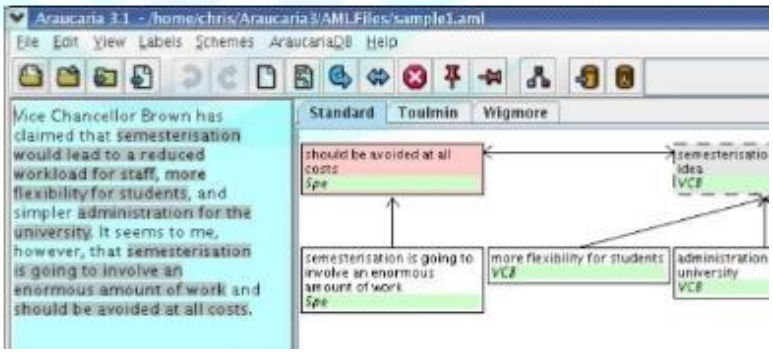


Figure 1. Example Sense-making tool: Araucaria [9]

However, these sense-making tools focus on assisting the user in visualizing and evaluating arguments after the relevant components (e.g., argument and evidence) are extracted and appropriately linked. The transition process from a raw text document to an argument structure is left to the individual user. Argument structuring (or mapping) involves complex thought processes such as creating arguments, comprehending their logical connectivity, and analyzing

their weaknesses and strengths [10]. Despite its effectiveness in developing critical thinking skills, it has been generally considered impractical, especially on pen and paper [11]. To provide investigators a tool that is usable in practice, structuring arguments is a task that needs to be automated.

Therefore, in this research, we first focus on automatically extracting arguments in case-related documents and finding the relations between the arguments using argument mining methods.

Based on the extracted argument structure, we attempt to provide relevant alternative hypotheses by finding similar arguments. Decision making in criminal cases is generally understood as selecting the most probable, well-supported story [12], [13], so it is necessary to construct several stories to compare and evaluate.

The table below shows the several sub-goals we have set for each task.

Table 1. Tasks and Sub-Goals of the Research

Nr.	Task	Objective
1	Collecting and Annotating Data	Generate a dataset that has been annotated using a crime analysis model (gold standard dataset)
2	Identifying argument components	Find the best features and text classification method to identify argument components compared to the gold standard dataset
3	Grouping argument components	Find the best clustering method compared to the gold standard dataset
4	Building an alternative hypothesis generation system	Test and compare the results of proposed methods to find alternative hypotheses

3. Limitation of Research

The biggest limitation of this research is the data. While our primary aim is to propose a system that can help an investigator analyze and evaluate their cases, we did not have access to a sufficient number of police investigation reports due to legal restrictions. Thus, we use court decisions of the first instance (district level) criminal courts in Korea.

The Korean judicial system is a three instance trial system, in which the first two instances rule based on fact evaluation and legal application, while the Supreme court focuses only on the interpretation of the law. The count of the charge brought to the first instance court is the same as the original crime investigation report; in this aspect, the judge's role as evaluator can be considered the same as the role of the investigator who is analyzing case files.

In both crime investigation reports and first instance court decisions, two parties try to prove and assert their argument using supportive statements.

Based on these similarities, we believe that the methodology we develop using court decision documents can be used and applied to assist with crime report analysis in the future.

4. Thesis Outline

Chapter II is a literature review of argument representation and analysis in crime investigation and the usage of argument mining techniques in various fields, including the legal domain. We also explain the algorithms involved in the argument mining procedure and present related tools for argument mining and crime analysis.

Chapter III describes the necessity of building an annotated corpus for this study and the procedure taken, and the dataset's final analysis.

Chapter IV gives an overview of the model and describes each process in detail. For argument component detection, we use four different classifiers to automatically differentiate argumentative text from non-argumentative text and predict the categories of our argument components. For argument clustering, we use K-means and Fuzzy c-means clustering methods to gather argument components into meaningful argument groups. We use Doc2Vec to calculate the similarities between sentences, specific rules to find argument components that can serve as alternative hypotheses, and utilize other similarity measures to retrieve the most relevant alternative hypotheses to the query.

Chapter V shows the result of each process and its analysis. For argument component identification, we provide the performance scores of each classifier and an analysis of the misclassified data. We give a detailed comparative analysis between the clustering results based on the algorithm and the number of selected clusters. Lastly, we show the result of the alternative hypothesis retrieval model with a test query. We also discuss the limitations that we have encountered during the experiments.

II. Literature Review

1. Argumentation in Crime Analysis

1) Crime Analysis and Sense-making

Although several definitions of crime analysis exist throughout literature, most of them are in consensus that crime analysis is a systematic study of crime and other relevant information to assist various operations of law enforcement [14]-[17]. For example, crime analysis can be used as a tactical tool to compare and analyze case data to identify patterns, suspects and therefore prevent or reduce certain types of criminal activities; however, it can also be used in strategic planning, e.g., allocation of workforce and resources [17].
One of the more detailed definition was proposed by the International Association of Crime Analysts (IACA):

> *A profession and process in which a set of quantitative and qualitative techniques are used to analyze data valuable to police agencies and their communities. It includes the analysis of crime and criminals, crime victims, disorder, quality of life issues, traffic issues, and internal police operations, and its results support criminal investigation and prosecution, patrol activities, crime prevention and reduction strategies, problem-solving, and the evaluation of police efforts* [18].

Ever since the formation of the first modern police in the early 19th century, a growing number of researchers have focused on understanding the crime analysis process and its implementation strategies [14].

Rachel Boba Santos [14] explains that crime analysis is conducted in five major steps: data collection, data collation, analysis, dissemination of the analysis results, and feedback incorporation. Data collation refers to correcting the data and adding necessary variables. The analysis step includes another subcycle (called the "data modification subcycle") that leads the analyst to return to the data collection or collation step for improvement.

Another popular model of conceptualizing the process of analyzing crimes is the sensemaking loop model proposed by Pirolli and Card [3]. According to the authors, the cognitive task analysis consists of two major loops: the foraging loop and the sensemaking loop.

The foraging loop involves exploring to increase the set of information, narrowing it down to more relevant data, then finally reading and analyzing the documents. It is similar to the data collection and collation step explained by [14], as it focuses on information retrieval and evaluation rather than gaining insight into the information.

Sensemaking can be defined as the process of finding a representation to organize information that helps the analysts filter and interpret the data while continuously improving the adaptation of the information to the schema or reducing the cost of operations [19], [20]. The sensemaking loop in the model is mostly derived from the work of [19], who have attempted to identify and categorize the tasks involved in the process of sensemaking by analyzing the work process of an education team trying to create a generic training course on laser printers. Russel et al. (1993) have identified three main processes ("Learning Loop Complex"):

(1) The Generation Loop: Creating and searching for a representation method that can aid the information retrieval process.
(2) The Data Coverage Loop: Identifying pertinent information and encode it in the representation method.

(3) The Representational Shift Loop: Through the process of (2), data that do not fit or is missing is identified ("residue"). The schema is then expanded or modified to accommodate the residue data. This loop aims to reduce the operation cost, such as the time of the overall task, or bring other improvements.

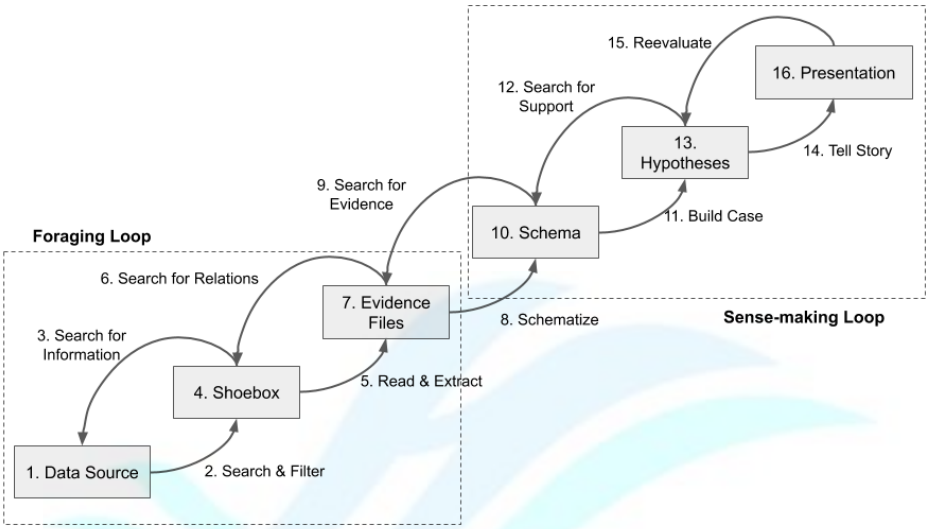


Figure 2. The Sensemaking Loop, adapted from [3]

Furthermore, Pirolli and Card [3] explain that certain leverage points in both foraging and sensemaking loops can occur during the analysis. A major time cost-related task in the foraging loop is scanning and finding relevant items from the data. This is also addressed in [19], which confirms that data extraction from documents is often the most time-consuming task. In their study, data extraction included finding relevant documents, selecting the related information sections of the documents, and encoding them on schemas form. In fact, 75% of the total time was spent extracting data and transforming them into the representation form [19].

By reducing the cost of a step, e.g., highlighting relevant information, offering summaries, analysts will be able to focus on other steps in the sensemaking

process, which can enhance their performance or capacity in general [3], [19].

Leverage points in the sensemaking loop are related to problem structuring, reasoning, and decision making. Generating hypotheses and letting them compete against each other is an essential part of the sensemaking process and preferable to testing the plausibility of one hypothesis individually [21]. However, time pressure and overload of data are detrimental to the analyst's ability to generate, manage, and evaluate hypotheses. Generating a set of alternative hypotheses to cover the space of possibility has been suggested to alleviate this concern [3].

2) Arguments and Argument Structures

An argument is a set of statements or premises linked with pieces of facts ("evidence") to support an idea, also referred to as a claim [10], [22], [23]. Argumentation can be described as the process where arguments are constructed, presented, interpreted, and evaluated to determine the claim's degree of truth [24]. It can also be referred to as understanding the method in which a conclusion or justification is established, i.e., the apodicticity in Aristotelean logic [25].

The claim can become a premise of another claim, which can be then with other elements, creating a chain of reasoning [22]. Argumentation plays a crucial role in many human reasoning focused domains such as the legal domain or academia; the ability to formulate a convincing argument is also crucial in decision making and analyzing other claims [26].

Analyzing arguments often includes identifying the components and determining the relationship between the components [5]. This leads to the development of several methods representing an argument using visual forms, sometimes also referred to as argument mapping [10], [27]. Most of the representation models use a diagram that can show the relations (e.g., support

or rebuttal) between the argument components, and mark inferred data from supporting facts or use graphic methods to represent conflict [5].

a. Argumentation Tree Diagram

One of the first diagrams representing "a train of arguments" was introduced by Richard Whately in the early 19th century [27]. Whately described his method as a *"convenient mode of exhibiting the logical analysis of a course of an argument, to draw it out in the form of a Tree, or Logical Division" (p.422);* to structure arguments to a form logical rules could be applied to [27]. To draw the diagram, Whately instructed first to identify the argument's conclusion, trace back the reasoning, and inspect the grounds the claim was made. The process should be repeated, using the grounds as claims to find further premises, forming a "*chain of arguments*".

In the 1950s, Beardsley proposed the first basic types of argument structures [28]. Arguments were divided into statements, which were represented as nodes, in the form of circled numbers. The link between the statements was expressed with the usage of arrows between the nodes [27].

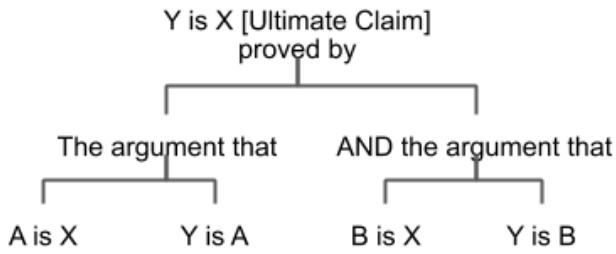


Figure 3. Whately's Diagram, adapted from [27]

Argumentation in tree structures is mostly used to show simple conclusion-premise relationships [22]. They usually have a root node ("top node") representing the main argument or conclusion of the structure. The premise

nodes are statements that can either support the argument independently or in combined form. Inferences are statements that serve as a logical bridge between premises or premise and conclusion. Nodes that represent rebuttals against the claim is commonly used as well.

b. Wigmore's Charting Method

This charting method was developed in the early 20th century by John Wigmore to teach his students how to analyze court decisions [29]. As this method's primary aim was to portray legal arguments, Wigmore's method focuses on classifying the argument components based on the role they play in the court case [30].

Wigmore acknowledges three types of evidence depending on the party, usually the defendant and prosecution [30]. The first is evidential data, which includes witness testimonies and circumstantial evidence. The second is corroborative data, which purpose is to support a claim or inference. The third is explanatory data, which explains the circumstantial evidence or reduces the witness' credibility. The evidence types are categorized once more depending on their role, e.g., testimonial evidence to support the prosecution's claim, testimonial evidence to rebut the prosecution's claim, testimonial evidence to support the defendant's claim, etc. This amounts to a total of 12 categories for evidence.

Wigmore's charting method is distinctive by its simple visual form; it does not use the entire text of the statement on the diagram. Instead, it uses shapes such as rectangles (testimonial evidence) and circles (other facts) to represent the facts of the case. The actual statement is collected in the evidence list (or key list), while the shapes hold the corresponding number to the original information piece. Lines are used to representing the accepted strength of the evidence (degree of belief). One arrow shows the direction of support, double arrows (the line between

Node 2 and 3,5,7,9) represent a strong supportive relationship.

Nodes with a > symbol (See node 11 in Figure 4) are explanatory evidence. Closed triangular nodes (Node 12, 13) are corroborative evidence.

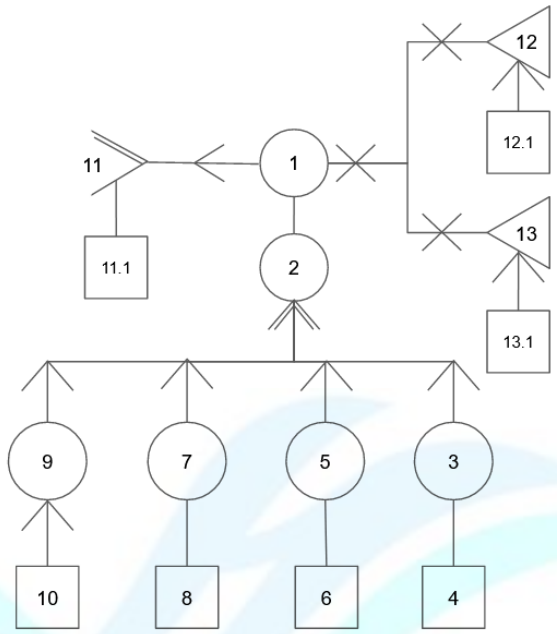


Figure 4. Wigmore's Charting Method, adapted from [29]

c. Toulmin's Argument Model

In his work *The Uses of Argument*, originally published in 1958, Toulmin proposed a new method to layout the elements of an argument [25]. His model was to be used primarily for jurisprudence [25] to analyze legal argumentation. Legal argumentation is distinguished by the fact that it aims to achieve justice, not simply focused on finding out the truth – which is the main purpose of ordinary argumentation [21]. Toulmin believed in a standardized form of the legal process, which resulted in a general pattern with some variants [25]. This pattern could be laid out using his model for argumentation.

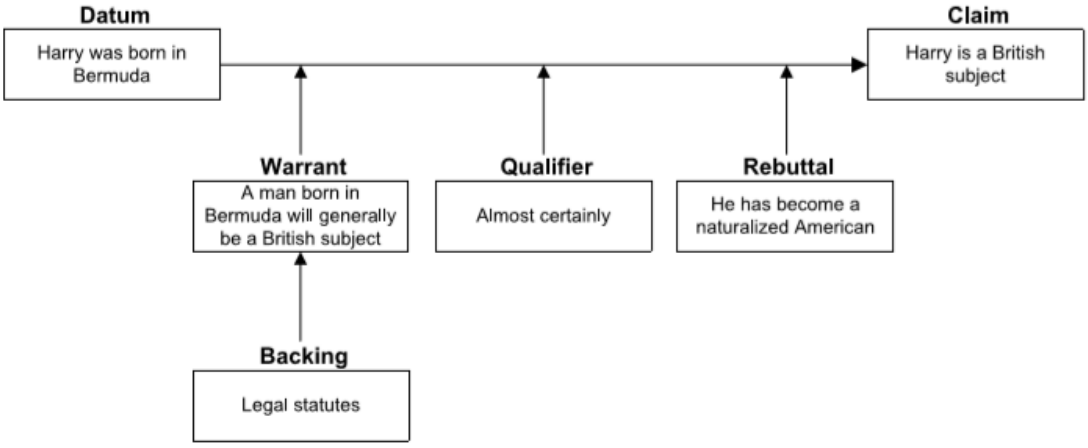


Figure 5. Toulmin's Argument Model Adapted from [25]

There are six argument components in Toulmin's model. The claim of the argument or the conclusion is the statement we wish to assert. The facts that lead to the claim is called a *datum*. The propositions that bridge the logical gap between datum and claim are referred to as a *warrant*. For example, if we want to establish "Harry is a British Subject (C)" from the data "Harry was born in Bermuda (D)," we need an inference statement such as "A man born in Bermuda will generally be a British Subject (W)" as a logical stepping stone. *Rebuttal* is a condition that could defeat the authority of the warranted conclusion. *Qualifiers* imply the "degree of force" the data supports the claim with the help of the warrant. *Backing* asserts the acceptability of the warrant. Whether a statement is a warrant or backing depends on its function; warrants serve as hypothetical, logical bridges, while backing can be in a statement of fact [25]. Backing and data can be distinguished by their role; backing aims to give authority to warrants while data attempts to support the claim.

2. Argument Mining

1) Argument Mining in Literature

In logic, argumentation is often represented in symbolic language that can be used to determine the logical strength of an argument [31]. This makes it easier to apply rules of argumentation or rules of logic to analyze the statements. However, in many areas that utilize human reasoning, such as journalism, legal practice, or academia, arguments are implied or inferred and cannot be expressed in purely formal representation [22]. As we have seen in the case study in 1. 1), the task of finding and selecting relevant information is fundamental and most time-consuming in the process of argumentation analysis. To alleviate the strain of the argumentation analysis process, the concept of argument mining has appeared around 2010 [32].

Argument mining is a research area that utilizes natural language processing and other knowledge representation and reasoning techniques based on linguistic, formal argumentation theories [22], [33].

The purpose of argument mining is to automatically identify argumentation elements and their structure in the document[34]. Through argument mining, researchers not only does attempt to differentiate arguments from non-arguments in documents [22], they also aim to categorize individual statements in the argumentation (e.g., premise and conclusion) and detect the relationship between the components [35].

One thing to note is that argumentation mining by itself does not provide the correctness or validity of arguments [22]. However, collecting arguments and their structure can help not only to understand the logical flow of the document, but analysts can make use of supporting tools (such as visualization) for easier interpretation in complex cases, and also compare similar arguments and their argumentative structure to extract patterns [32].

While argument mining combines different methods and theories from diverse research areas, there are two core tasks in the process:

1) Argument extraction: Detecting and identifying arguments from the natural language text.

This task is similar to finding and selecting relevant information sections in the documents in the sensemaking process. Most approaches in the literature [22], [23], [36], [37] suggest machine learning methods such as Support Vector Machines (SVM), Naïve Bayes, or Logistic Regression on annotated data to detect argument components and identify their role.

2) Argument structure construction: The automatic reconstruction of the extracted argument components.

This task aims to identify which arguments are related and what that relationship represents, e.g., support or attack. Throughout literature, several researchers have used diverse methods for this task, from supervised SVM, Naïve Bayes, Logistic Regression [35], unsupervised clustering methods [38], and text entailment [39]. The predicted output can be visualized in argument representation forms.

Argument mining has been applied to several domains and domain-specific datasets. An overview of recent literature and its purposes are shown in the figure below.

Table 2. Argument Mining in Literature Overview

Domain	Objective	Research
Education	Identification of arguments in persuasive essays and improving the performance of their automatic scoring system.	[26]
Academia	Analysis of the structure of scientific articles by dividing them into zones (Argumentative zoning).	[40]
Journalism	Providing news summary, current trends, and customizing options for users. *[41] also claims that advanced argument mining can be used to detect fake news.	[37]
Social media	Analysis of arguments in social media text to identify users' opinions about products or policies that can support the decision-making process.	[42]
Policy	Augmenting comprehension by providing analysis and visualization of arguments in policy discussions.	[36]

2) Argument Mining for Legal Documents

In the legal domain, especially when handling legal documents, legal argumentation is the prime focus [21, p. 30]. In fact, legal drafts have been encouraged to follow modern logic to improve the precision of the legal language, i.e., removing the uncertainty the occurs due to omitting of facts or uncertainty caused by the written statement itself [43].

The interest of using argument-assistance systems for the legal profession is not new; The authors of [6] developed an argument visualization system to assist lawyers in structuring their arguments when drafting pleadings to the court. Other arguments can be found in legislative texts, case law, and doctrinal text that proposes a specific interpretation of a legal norm [44].

In a study regarding detecting legal argumentation, the researchers in [44] have used a multinomial naïve Bayes classifier and a maximum entropy model on the Araucaria dataset, which consists of arguments from diverse sources. The test result showed out of all sources, arguments in newspapers were the easiest to detect (accuracy rate of 76%) while arguments in legal judgment scored the lowest accuracy rate (65%). The authors state that this could be due to the small dataset and the more complex argumentation pattern.

Mochales and Moens [22] experimented with legal text from the European Court of Human Rights (ECHR) to detect argumentation and its structure. They first used features such as n-gram, verbs, adjectives, punctuation to classify argumentative from the non-argumentative text, then argumentative patterns to detect premises and conclusion from the argumentative data. To identify the argument structure, the authors have composed a simple context-free grammar (CFG) to parse the text, which reached a 60% accuracy rate. This study showed that classifying arguments from non-arguments in legal documents is possible.

A more recent study was conducted by [38]. This research was also conducted on the ECHR corpus. The authors proposed a system to detect premise and conclusion using classifiers and used fuzzy c-means clustering to group relevant argument elements together. The authors also developed the "Appropriate Cluster Identification Algorithm" to evaluate the clusters against the human-defined gold-standard cluster. Their clustering method to identify legal arguments reached an average accuracy of 59%.

3. Argument Mining Methodology

As explained in the previous section, argument mining is a combined approach to detect, extract, and reconstruct arguments from natural text. This section will discuss the theoretical concepts and algorithms used in previous research for each step in the argument mining process.

1) Argumentative Sentence Detection

According to [44], argument detection is a text classification problem. A classifier can be trained on the annotated data to detect arguments automatically and classify their type [44]. Usually, argument detection is divided into two sub-process [32]. First, classify argument-relevant data and non-argumentative data. Second, classify the identified argumentative data. This procedure is concurrent with the annotation procedure we have developed.

Previous research has shown that the first sub-process is relatively simple and has a high success rate [22], [45]. In contrast, classifying the type of argument could be a challenge [26].

To classify text using classification algorithms, features of the data need to be extracted.

a. Feature Extraction

Features in natural language processing refer to numeric or symbolic values representing the sentence and can be used as input data for classifiers [45], [46]. Based on previous text classification researches [22], [42], [44], [46], [47], features such as n-gram or POS tagging are commonly used.

N-gram is a sequence of 1 to n successive tokens (words) from the sentence. For example, unigram (n = 1) is each word in the sentence.

Table 3. Example of N-gram

Unigram (n=1)	Bigram (n=2)	Trigram (n=3)
“The”,”quick”,”brown”, ”fox”,”jumped”…	“The quick”,”quick brown”, ”brown fox”…	“The quick brown”, “quick brown fox”, ”brown fox jumped”…

The words are normalized with the Term Frequency – Inverse Document Frequency (TF-IDF) method. TF-IDF evaluates the relevance of a word in a document. Term frequency refers to the number of times a word has occurred in the document (tf), while document frequency is the fraction of documents the word has occurred, inversed:

$$TF - IDF = tf \cdot log\left(\frac{D}{D_w}\right)$$

where D is the total number of documents and D_w is the number of documents the word occurs. The weighted terms can be used to find important keywords representing the document – a higher TF-IDF value implies less frequently used words.

Part of Speech (POS) tagging refers to the process of identifying morphemes by their definition and context of the sentence. For example:

>> 피해자는 질식사하였다고 볼 수 없다.
>> 피해자/NNG + 는/JX + 질식사/NNG + 하/XSV + 았/EP + 다고/EC + 보/VV + ㄹ/ETM + 수/NNB + 없/VA + 다/EF + ./SF

The categorization of POS depends on the analyzer. The most popular python wrapper for Korean natural language processing is called KoNLPy and offers 5 POS morphological analyzers (Kkma, Hannanum, Komoran, Okt, Mecab)[4]. Each

[4] https://konlpy.org/

tagger has its strength and issues; therefore, it is important to use the appropriate tagger depending on the purpose.

b. Classification Algorithm

Several machine learning algorithms were used in literature to classify argumentative text and non-argumentative text. According to Cabrio and Villata [32], Support Vector Machine (SVM) and Logistic Regression are among the most used sentence classification algorithms.

Previous researchers analyzing legal document data also used the Naïve Bayes classifier [22], [44].

(1) Naïve Bayes

The Naive Bayes classifier is a commonly used supervised learning method. The model calculates the most probable outcome using joint probability, which is calculated based on Bayes' conditional probability. The Bayes' theorem can be expressed as follows:

$$P(A|B) = \frac{P(A)\,P(B|A)}{P(B)}$$

where $P(A|B)$ is the likelihood of an event, A occurring given event B is true. $P(B|A)$ is the likelihood of event B occurring given A is true. $P(A)$ and $P(B)$ are the probability of respective observations. The joint model of a sentence s being in category c can be computed as:

$$P(c|s) \propto P(c)\prod_{1=1}^{n_s} P(x_i|c)$$

where $P(c|s)$ is proportional to the right side of the equation; $P(x_i|c)$ refers to the conditional probability of feature x occurring in a sentence, which is labeled as category c. This can be used to measure how much the evidence x_i contributes to c as the correct category [48, p. 258].

$\{x_1, x_2 \dots x_{n_s}\}$ refer to the vector of features, where n_s is the number of features found in s. $P(c)$ is the likelihood of a sentence occurring in c. Note that the equation above is simplified as the denominator $P(x_1, x_2, \dots x_{n_s})$ is constant in the input.

The model is considered naïve as it assumes conditional independence between all the pairs of features in a class. The classifier then selects the most probable outcome based on the *maximum a posterori* (MAP) decision rule.

$$\hat{c} = arg\max_{c} P(c) \prod_{i=1}^{n_s} P(x_i|c)$$

There are several variations of this model. This study used the Multinomial Naive Bayes classifier, which is often used for text classification purposes. In this classifier, the data is typically represented as the frequency of each feature, such as word vector counts. The parameters of the distribution are predicted by using a smoothed version of maximum likelihood. The smoothing parameter can also be used to avoid zero probabilities in computations.

(2) Logistic Regression

Logistic regression is a probabilistic method that classifies binary classes such as pass or fail, alive or dead. Each class's probability is assigned a value between 0 and 1, which sum results in 1. If multiple classes are given, the model uses the one-vs-rest scheme, which splits the dataset into multiple binary classification instances. Another option is to use a multinomial logistic regression model (also

known as the maximum entropy classifier).

As its name suggests, the logistic regression classifier bases its calculation on logistic function (sigmoid curve):

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

where L is the curve's maximum value, x_0 is the midpoint of the sigmoid curve, k refers to the steepness of the curve.

The equation for predicting the probability of class (y) being equal to 1 given the feature set x and parameterized by θ, can be expressed as [49, p. 49]:

$$P(y = 1 \mid x; \theta) = \frac{1}{1 + e^{-\theta^T x}}$$
$$P(y = 0 \mid x; \theta) = 1 - \frac{1}{1 + e^{-\theta^T x}}$$

in which θ^T is the transposed matrix of a vector of parameters. The predicted probabilities are fitted to classes using the likelihood function.

$$L(\theta) = \prod_{i=1}^{n} P(x_i)^{y_i} (1 - P(x_i))^{1-y_i}$$

where x_i is a vector of features and y_i are the classes observed [50, p. 227]. The log-likelihood is the actual cost function of the logistic regression[50, p. 228]:

$$LL(\theta) = \sum_{i=1}^{n} y_i \, logP(x_i) + (1 - y_i) log(1 - P(x_i))$$

The maximum likelihood estimation computes the log-likelihood and finds the values of θ that maximize the outcome. The negative log-likelihood ($-LL(\theta)$) is used as a cost function for the model. In machine learning, the cost function is

often regularized. Regularization refers to penalties applied to large weight coefficients in the model by adding additional values to the cost function. This helps the model to prevent complexity and avoids overfitting issues.

(3) Support Vector Machine (SVM)

The SVM model is another popular supervised learning methods for classification. This model uses a line called hyperplane (bold line in Figure 6) to separate the data into classes (filled circles and triangles).

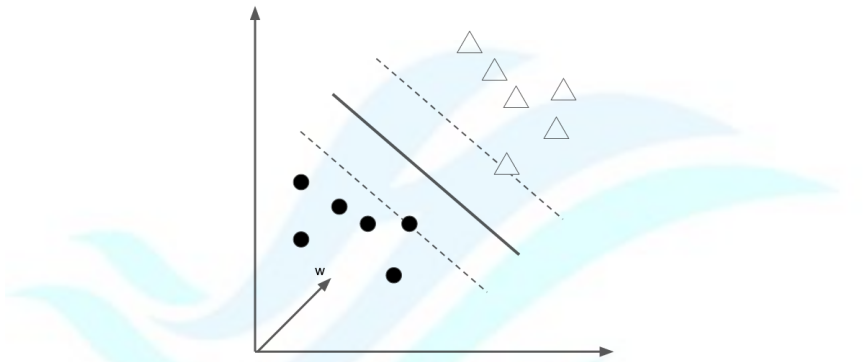


Figure 6. SVM Classifier Visualization

The distance (d) between the classes and the hyperplane is called margin, while the data on the margin are called support vectors. An optimal hyperplane refers to a line that maximizes the distance between the closest points in all the classes.

The equation of the hyperplane where x is a p-dimensional vector is expressed as:

$$w \cdot x + b = 0$$

where w is the direction or weight vector (arrow in Figure 6) and b is the bias. The function of the classifier can be defined as:

$$y = +1, \ when\, w \cdot x + b \geq 0$$

$$y = -1, \ when\, w \cdot x + b < 0$$

Thus, all the data above or on the hyperplane will be labeled (y) as class +1, while the data points below will be categorized as class -1.

SVMs are primarily classifiers for binary classes. To work with multiple classes, the one-vs-rest scheme can be implemented. The class that produces the largest margin or the class chosen by most classifiers can be selected as the final classification [48, p. 330].

2) Argument Reconstruction

To fully grasp the argument's meaning and reconstruct the sentences accordingly, the structure and relation between the argumentative texts must be detected. In their work [46], Wyner et al. suggest context-free grammar schemes find matching patterns in text. Context-free grammar use rules to identify important markers, e.g., "therefore" is a conclusive marker while "however" is a contrast marker, to analyze the structure of the argumentative sentences. The premise is that the grammatical of legal documents share similar constructs that can be expressed to a set of rules. The limitation of this approach is when sentences use an uncommon structure.

Another method using machine learning algorithms on annotated argument relationships was suggested by Lawrence and Reed [51]. In their work, the authors have identified four types of argumentation schemes in their datasets, such as analogy, case to effect, practical reasoning, and verbal classification. One-vs-all classifiers were used to classify the type of proposition in the data. However, the focus of this research is to recognize certain argumentation schemes, which our annotated data does not reflect.

In his doctorate dissertation, [45] used unsupervised clustering methods to group argumentative sentences into clusters. The base hypothesis of using clustering methods on documents is that the related documents share similarities, for example, in semantics features [48, p. 350].

This research focuses on the document clustering approach proposed by Poudyal [45] to group argumentative sentences.

a. Feature Selection

To apply the clustering methods to the documents, the words need to be vectorized. One of the most popular methods to vectorize words is Word2Vec. Sentence closeness was included as an additional feature.

(1) Word2Vec

Word2Vec refers to a group of related algorithms that can distribute words to be represented in a vector space, i.e., word embedding. It was first introduced in the papers [52] and [53] and is known to perform better than previous models that compute the word representations, such as Latent Semantic Analysis. The core premise of the model is that similar words are not only close to each other but also "multiple degrees of similarity"[52].

Word2Vec is known for two types of architectures: the Continuous Bag-of-Words model and Skip-gram model. A visual representation of the two models is shown below.

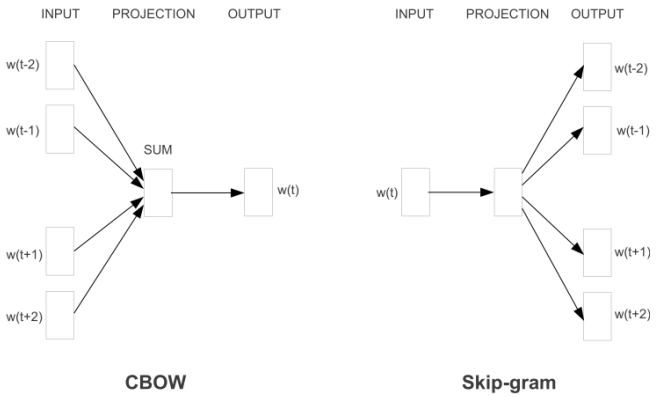


Figure 7. CBOW and Skip-gram Model [52]

The Word2Vec model consists of three layers: the input, the hidden

(projection), and the output layer. The inputs are words with their weights calculated depending on the distance to the current word. The outputs are the embedding vectors.

The difference between the two predictive architecture CBOW and Skip-gram is whether the target word is input or output. The CBOW model uses words in history and future (context words) to predict the target word. Skip-gram, on the other hand, uses the target word to predict the context words. CBOW is known to be faster to train than Skip-gram models and achieves slightly better accuracy on frequent words. Skip-gram works better on smaller datasets and works better with rare words compared to CBOW.

Context windows are used to define the context words that are to be used. Its size refers to the distance between the target word and the neighboring context word [54]. Selecting the right context window size is significant in finding the most appropriate word representation.

(2) Sentence Closeness

Sentence closeness refers to the distance between a sentence and other sentences in a document. As most sentences in the same argument are closely located, this could be a useful feature to determine argument groups.

For sentence 1(s1) and 2 (s2), sentence closeness is calculated as suggested by [38] :

$$Closeness(s_1, s_2) = \frac{1}{1 + |n(s_1) - n(s_2)|}$$

n refers to the position number of the sentence within the document. The sentence closeness of the same sentences will be 1; greater value implies closer sentences.

b. Clustering Algorithm

Several clustering methods have been developed and researched throughout the years, depending on the purpose and data [55]. K-means and agglomerative hierarchical clustering are usually considered good approaches for document clustering [56, p. 2]. However, Steinbach et al. [56, p. 16] also state that agglomerative hierarchical clustering shows poor performance when the nearest neighbors are unreliable. Like their data, we use all documents' vocabulary, which can lead to documents to be identified as nearest neighbors, although they are in different classes.

Another clustering method suggested by [38] to cluster arguments in legal documents is Fuzzy c-means.

Therefore, we look into K-means and Fuzzy c-means as our clustering methods.

(1) K-means

K-means is a form of clustering algorithm that partitions n observations into k clusters using the nearest cluster centroids – which are the mean value of the data points within a cluster. The core process of k-means works as presented below[57]:

① Randomly select k data points as cluster centroids (prototype).
② Compute each data point's similarity to each cluster centroid and assign all points to the nearest centroid.
③ Update the k centroids of each cluster.
④ Repeat steps 2 and 3 until centroids do not change between iterations.

Other termination conditions can be the completion of a fixed number of

iterations, document assignment to clusters are fixed between iterations, the residual sum of squares (used to represent how well centroids represent the data points within the respective clusters) is below threshold [48, p. 360]

K-means is a hard clustering algorithm as the data points are assigned to only one cluster. The objective of K-means is to minimize:

$$arg\min_{k} \sum_{j=1}^{k} \sum_{i=1}^{N} \left\| x_i^{(j)} - \mu_j \right\|^2$$

where N is the number of data points and k is the number of clusters; x_i refers to each data point while μ_j is the mean of the data points in cluster j. $\left\| x_i^{(j)} - \mu_j \right\|^2$ calculates the euclidean distance between the two points.

(2) Fuzzy c-means

Fuzzy logic is a form of logic where membership in a fuzzy set is expressed in degrees of truth[58]. This makes it possible to apply logic not only to data with bivalent values such as "old" or "young" but also to granular values, such as "not very young"[59, p. 2754]. In Zadeh's work, it is further explained that fuzzy logic (as opposed to bivalent logic) is ideal for computing human perceptions due to its tolerance for imprecision and approximation [59, p. 2770].

The same logic is used in fuzzy clustering. Fuzzy c-means is a soft clustering method and permits data points to be assigned to more than one cluster.

The process of FCM is similar to K-means: assign the data points to the clusters and repeat the process until convergence is reached. However, whereas K-means assigned each data point a crisp cluster label, data points in FCM have a membership in each cluster center, expressed as a percentage value between 0 to 100 percent. The similarity can also be seen in the equation itself.

The FCM algorithm aims to minimize the objective function, which can be

described as:

$$arg\min_{C} \sum_{j=1}^{C} \sum_{i=1}^{N} w_{ij}^{m} \|x_i - c_j\|^2$$

where N is the number of data points and $\{x_1, x_2 \ldots x_N\}$ is are the collection of data points; C is the number of clusters and $\{c_1, c_2, .., c_C\}$ represent the cluster centers of each cluster. m refers to the fuzzifier parameter and w_ij is the membership degrees of x_i belonging to the j-th cluster (c_j).

3) Alternative Hypothesis Generation

While generating or inventing hypotheses is not a necessary step in the argument mining process, it is crucial to analyze arguments. In the work of Reed and Rowe [9], the researchers have developed a tool they have named Araucaria (version 3.1, the predecessor of OVA+, an online tool for argument analysis), which provides the user two types of suggestions to build alternative hypotheses [9]:

- In the form of a critical question
- In the form of opposing argument

The critical questions were pre-determined and provided the user thinking steps to assess their arguments (See Figure 8) critically.

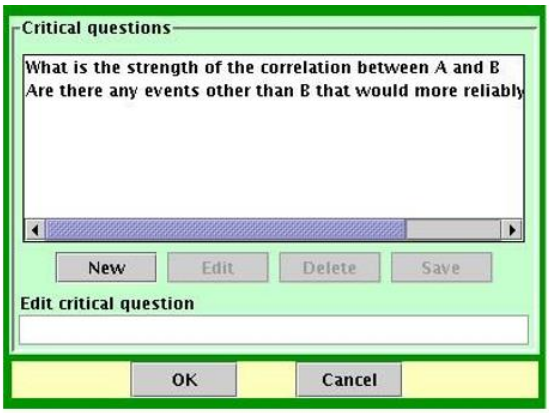


Figure 8. Araucaria's Scheme Edit Window (Critical Question)

The latter function was provided as a window (The scheme edit window) to create premises and conclusions that could be evaluated based on logic rulesets (See Figure 9).

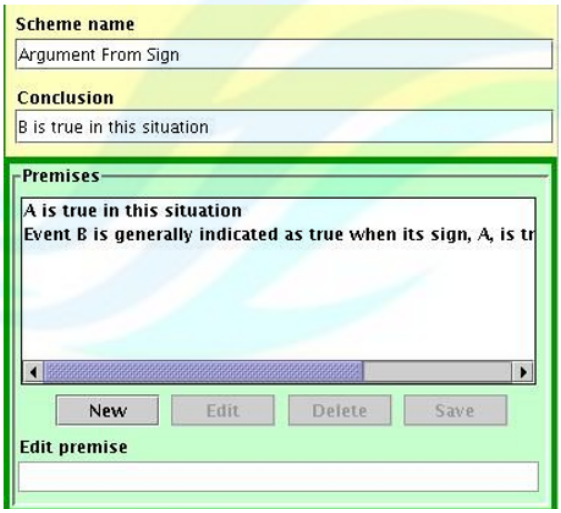


Figure 9. Araucaria's Scheme Edit Window (Premise Conclusion Edit)

Hypothesis generation in crime investigations is mostly dependant on the investigator's personal expertise. This usually consists of domain knowledge and experience, which leads to a discrepancy of the hypothesis generation capacity between an experienced officer and less experienced investigators [60].

A method to support investigators in the hypothesis generation process is finding similar cases. Several systems were developed to serve this purpose (See Section II. 4). By finding similar cases and comparing them, the investigator can

consider different scenarios and create more hypotheses or expect a potential outcome.

In this research, we focus on finding similar arguments rather than finding similar cases and attempt to retrieve data that can serve as alternative hypotheses based on the component. To find the most similar and relevant sentences in our database, we use similarity values based on Doc2Vec and graph similarity.

(1) Doc2Vec

Doc2Vec was first introduced in [61]. It is similar to Word2Vec with the difference that it attempts to capture the relationship between the documents (or sentences, paragraphs), whereas Word2Vec focuses on finding the relationship between words. Inspired from Word2Vec, two architectures are proposed to predict either the target word or the context words:

Paragraph Vector – Distributed Memory (PV-DM) takes context words and a paragraph, which is represented as a paragraph matrix. A paragraph matrix contains the vectors of the paragraph (i.e., fixed-length feature representations that can be applied to texts in various lengths [61]) that can also be used to keep the topic of the paragraph. The input is then concatenated and used to predict the output with a multiclass classifier such as softmax.

Paragraph Vector – Distributed Bag of Words (PV-DBOW), on the other hand, ignores the context words in the input layer and tries to predict them in the output layer. The paragraph vector serves as the input, which is trained using a small window. This model is considered similar to the Skip-gram model used in Word2Vec.

(2) Graph Similarity

Graph similarity has been used to identify the text similarities between argument graphs in this study [62]. Their goal was to compute the similarities between argument graphs (consisting of information nodes and scheme nodes connected with arrows, i.e., edges) and provide support for the reasoning process [62, p. 221]. The concept of argument graph similarity is also introduced in the author's previous work [63] to enable retrieving similar workflows from the repository.

Graph similarity is the normalized sum of the similarities between nodes and similarities between edges. The nodes refer to the text data that represent components of the argument, while edges refer to the relationship between the nodes. The node similarity computes the similarity of nodes that are in the same category (or types). Node similarity between nodes in other categories is set to 0. Edge similarity is the average score of the similarities between the endpoints. High edge similarity indicates that similar nodes are connected through the link [63, p. 121]. Edge similarity following the interpretation of [62, p. 220] is calculated as:

$$similarity_E(e_q, e_c) = 0.5 \cdot (similarity(e_{ql}, e_{cl})) + similarity(e_{qr}, e_{cr}))$$

For the endpoints of the query graph (q) and the to-be-compared graph (c), the similarity of the nodes on the left side and the similarity of the nodes on the right side are computed (See Figure 10).

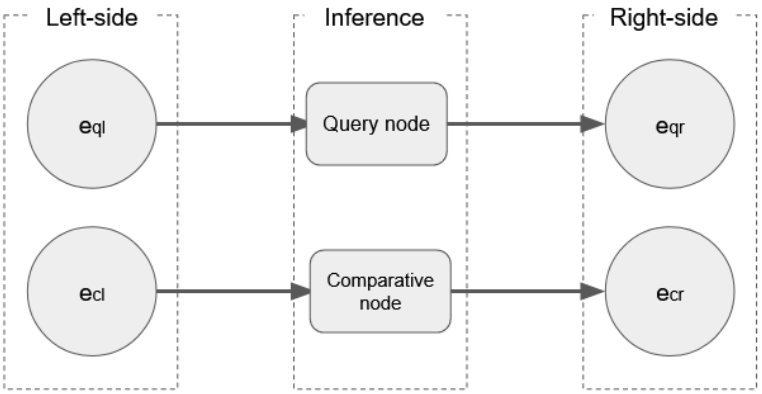


Figure 10. Edge Similarity Visualization

A standard cosine similarity can be used to calculate the similarity between the text data in the nodes.

4. Argument Mining and Crime Analysis Tools

In this section, we present tools that are related to our research. First, we discuss argument mining tools used to find relevant arguments or help the user create new arguments. Second, we look into crime analysis tools that assist in the foraging loop that can be used to generate hypotheses.

1) Argument Mining Tools

In contrast to argument visualization tools, argument mining tools focus on automatically suggesting potential arguments relevant to the topic. A prime example of one such tool is IBM's Debater project [64]. The Debater project has been introduced in 2014, showing how it can hold a debate with a human contestant. It utilizes text processing technology to find context dependant claims (CDC) in relevant Wikipedia articles [65]. The CDC is categorized as either pro- or con-argument based on their sentiment, which is then presented to the

audience in speech format using Text-To-Speech (TTS).

The Carneades Argumentation System, on the other hand, is an argument evaluation and invention tool [66]. The developers of Carneades assume that an argument mining system such as Debater builds a knowledge base in the background, which Carneades then can use to visualize and calculate which arguments need to be more backing to become acceptable. The Carneades argument assistant can apply pre-defined argument schemes to the premises in the knowledge base and help to generate (invent) new arguments.

Expert systems also have been utilized to help users with creating new scenarios. Keppens et al. [67] introduced an expert system that can visualize crime scenarios and the logical assumptions between claims. The system deconstructs event components of a crime scenario and automatically makes potential hypotheses based on the given evidence. The authors have built an Assumption Based Truth Maintenance System (ATMS) that keeps track of the plausibility of multiple hypotheses (crime scenarios) using abductive reasoning and inference. By analyzing and comparing the constructed crime scenarios, the system can give suggestions for further investigations [67].

2) Crime Analysis Tools

With the advancements in technology, several tools have been developed to support human investigators in their analysis process. The tools assist in several aspects of crime investigation, from crime pattern recognition to crime prediction [14, pp. 365-366].

The tools related to this research are systems that can provide the investigator similar cases to their current one, enabling them to generate potential hypotheses of suspects, crime patterns, or crime locations.

One such tool is Patternizr, which was jointly developed by the New York Police and IBM [68]. It seeks to alleviate police officers from solely relying on manual

and memory-based pattern identification and offer support by automating the process. Patternizr learned 10,000 patterns of robbery incidents in New York State over the past ten years, consisting of data such as date, time, location, crime subcategory, M.O., and suspect information [68, p. 2]. The similarity between cases is calculated with a random forest model. The system provides a map with a list of similar crimes found in the New York Police Department database from 2016 and a map that visualizes similar crime locations.

A system that has been developed in Korea is the Crime Layout Understanding Engine (CLUE). This system recommends similar cases by extracting important crime facts from the investigation result report entered in the Criminal Justice Portal (KICS) [2].

III. Text Annotation and Corpus Analysis

1. The Necessity of Building a New Corpus

There has been a great demand for more text datasets throughout the years that include labels of argument components and their relations [69].

One of the most known argument datasets is the Araucaria corpus collected and published by a research team at the University of Dundee [28]. The Araucaria corpus is a set of arguments from 19 newspapers, 4 parliamentary records, 5 court reports, 6 magazines, and 14 online discussion boards that have been collected in 2003.

The ECHR corpus is a popular dataset to conduct argument mining for legal purposes created by [22], [38], [44], [46]. It consists of decision case-law documents (average word length 3500) and judgments case-law documents (average word length 10000 words) from the European Court of Human Rights. In total, 47 documents were annotated[5].

To build the corpus, Mochales and Moens hired two lawyers to annotate the case-law documents following a guideline that described the arguments. Another lawyer was then selected to analyze the annotation and find the reason for discrepancies between the first two annotators. Based on the finding, a new guideline was created, and a fourth lawyer was hired to annotate and solve disagreements. In the end, the final inter-rater agreement was 75% using Cohen's kappa coefficient [22].

---

[5] In his doctoral dissertation [45], Poudyal also uses the ECHR corpus to extract legal argumentation. However, the total number of documents used in his research was 42 out of 43 (1 was omitted due to langague issue).

Neither of these corpora fits the goal of the study. To build a system that can be used by Korean investigators, the corpus needs to be in the Korean language, handling case documents from the Korean legal system.

To our knowledge, there is no annotated corpus that includes argument components and their relations in Korean legal documents. Research on legal documents in Korean mostly use text mining technologies and focus on finding relevant legal clauses [70] or extracting keywords ("Who, Where, When, What") [71].

Thus, before developing an argument mining system, it was necessary to generate our own corpus.

2. Building the Corpus

1) Collecting the Source Data

A pre-study of the source data must be conducted to understand the argumentation structure, which is crucial in determining an appropriate argument mining method. Especially in a verdict, understanding the argumental structure means understanding the legal debate, which can analyze the deciding factors in a case, lay out the logical flow for evaluation, and simply serve as a quick overview of the most important legal issue.

On January 1st, 2019, the Korean court started the Online Access to Court Records system (판결서 인터넷 열람 서비스) that enables access to all criminal court decision from January 1st, 2013 and other decision from January 1st, 2015 that was anonymized [72]. Only Supreme Court decisions were made public before these changes, while lower court judgments were only accessible by relevant parties. Critics have argued this limitation as a violation of the fundamental right to know and a public trial principle. The new system greatly improved accessibility, especially for criminal cases, as it enables search using

keywords. However, the new system as a data source is still limited as it only provides the data in PDF format and charges 1,000 KRW per case as a fee; only a maximum number of 5 cases can be selected per payment [72].

Other sources that can be used to collect court decisions are private search sites such as CaseNote[6] and LegalSearch[7]. While not all decisions are available, these search sites offer useful filter and joined-keyword search features, which are necessary to adjust the scope of the documents' type and topic. They are also shown in HTML format, making it easier to collect the data with a web-scraping tool.

For this study, we used a simple Python script to retrieve 100 first instance criminal court cases using the keywords: homicide (살인) and evaluation of (the defendant's) claim (주장에 대한 판단). The data was saved into text files. To find documents that share similarities to crime investigation reports, we set up the following criteria for data selection:

1) The main debate point of the decision is homicide (Act 250 of Korean Criminal Law)
2) The decision must contain a defense statement and a judges evaluation of that statement
3) The defense statements and judges evaluation must be in full sentences (not bullet points)

Out of the collected cases, we have manually omitted cases that were falsely matched or did not meet the criteria. In the end, we used a total of 73 cases to build the corpus.

[6] https://casenote.kr

[7] https://legalsearch.kr

2) Document Analysis

a. The Structure of the Court Decision

When analyzing the documents, it became clear that the documents were following a format to categorize the data (See table below).

Metadata	[사건] (Case number)
	피고인 (Defendant)
	검사 (Prosecution)
	변호인 (Defense)
	[배심원] (Jury)
	판결선고 (Date of Pronouncement)
Conclusion	주문 (Adjudication)
이유 (Ground for Decision)	
Details	범죄사실 (Criminal Fact)
	증거의 요지 (Essential Evidence List)
	법령의 적용 (Legal clause application)
	피고인 및 변호인의 주장에 관한 판단 (Evaluation of Defense)
	양형의 이유 (Ground for punishment)
Metadata	재판장 (presiding judge)

Figure 11. Structure of Korean Court Decisions

While the wordings can change, most of the observed verdicts followed this structure to formulate their judgment. The decision starts with metadata regarding the case, including the case number, date, and personal information about the relevant parties. Adjudication is mostly a one-line sentence that summarizes the verdict's outcome, i.e., guilty or not guilty. The details of the case are listed under the Ground for Decision (이유). The criminal fact is usually formulated as a narrative story by the prosecutor. Facts or arguments that support the defendant's claim are mostly discussed in the Evaluation of Defense (피고인 및 변호인의 주장에 관한 판단) or Ground for punishment (양형 이유).

Sentences in Essential Evidence List (증거의 요지) and Applicable Legal Clauses (법령의 적용) are mostly a list of bullet points.

Thus, most of the core arguments were found in the section Evaluation of Defense, starting with the defense's argument and the premises and conclusion to accept or deny it.

The section Ground for punishment also contained the judge's reasoning on what factors were decisive to support or dispute the verdict's conclusion. For this study, we mainly focus on the evaluation section of the decision.

b. Preprocessing

Before moving onto annotation, certain preprocessing steps were necessary to provide a uniform dataset.

(1) Basic Preprocessing

In the dataset, witness and expert testimonies were often quoted using quotation marks. A simple regular expression script was used to transform all quotation marks within the document into single quotation marks to unify these marks. Whitespace lines between sentences were also removed.

(2) Sentence Splitting

Argument mining is usually performed taking sentences as argument units [22], [34], [73]. A sentence can be defined as a set of words that conveys meaning, often consisting of one or more clauses [74]. The end of a sentence in a judgment is usually declared with a period mark (or full stop mark). However, whether the sentence-based approach is also applicable to Korean legal documents is debatable. Most of the observed sentences in the collected data were complex sentences that contained multiple premises and conclusions.

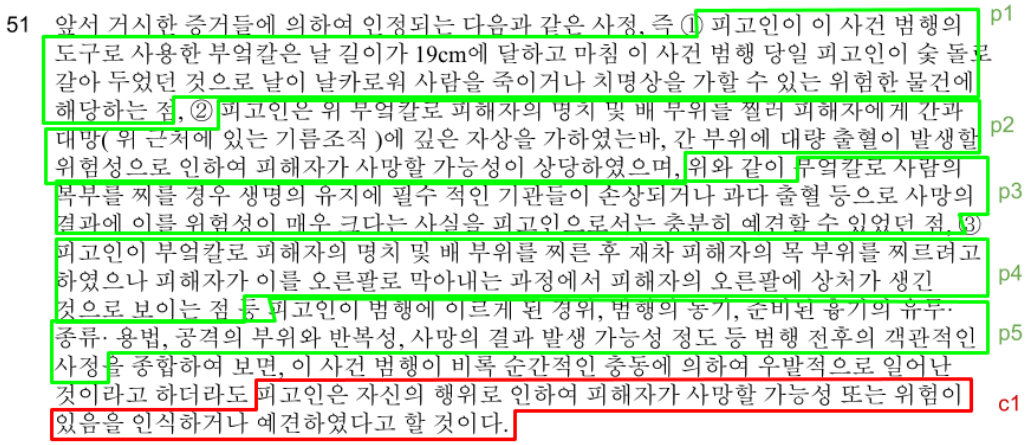


Figure 12. Example Complex sentence (2014고합441) [8]

Figure 7 shows an example of a sentence taken from a homicide case with multiple premises and conclusions. The green boxes show the premises (p1~p5) of the argument, while the red box shows the argument's conclusion. With sentences that are as complex as the given example, simply allowing multi-labeling (several labels assigned to the same sentence) is not enough – it would defeat the purpose of trying to understand and evaluate the argument by

[8] English translation: "*The following circumstances are acknowledged by the above stated evidence, namely, ① the kitchen knife that the defendant used as a tool for the crime of this case reached 19cm in length, and the blade was sharp as the defendant had sharpened it with whetstone on the day of the crime of this case and can be considered as a dangerous object that can kill or inflict fatal injuries, ② the defendant used the aforementioned kitchen knife to inflict a deep cut on the liver and gastrocolic omentum (fat tissue near the stomach) to the victim, the risk of death due to excessive bleeding or damage to organs was very high and the defendant was able to predict that stabbing a person with the aforementioned knife can cause damage to life essential organs or excessive bleeding leading to high risk of death, ③ there appears to be a wound on the right arm of the victim when the victim used his right arm to block the defendant's renewed attempt to stab the victim's neck again after piercing the victim's solar plexus and stomach with the kitchen knife, considering all the objective circumstances before and after the crime, such as how the events that led to the crime, the motive of the crime, the presence, type, and usage of the prepared weapons, the location and repetition of the attack, and the possibility of death as a result, even if the crime had occurred due to sudden impulse as voluntary manslaughter, the defendant recognized or predicted that there was a possibility or danger of the victim's death due to his actions.*"

automatically analyzing elements of arguments.

This is also a known problem in some of the sentences in the ECHR corpus [45]. Out of 2160 argumentative sentences, 254 sentences contained premises and conclusions within the sentence [45]. To differentiate the components within the sentence, a list of keywords such as 'that', 'because', 'and', 'since' or punctuation marks including commas and semicolons were used.

We have attempted several methods to split the sentence into useful phrases.

- *Automatic sentence detection using Kkma analyzer*

Kkma is a morphological analyzer and natural language processing system for Korean developed by the Intelligent Data System (IDS) laboratory at Seoul University. The system offers a sentence detection feature. Using the example sentence from Figure 13, the package splits it into two parts. This is most likely due to the fact the analyzer recognized a terminating end of a sentence (EFN) at the end of line 0 ("다고") and end of line 1 ("다").

0:앞서 거시한 증거들에 의하여 인정되는 다음과 같은 사정, 즉 ① 피고인이 이 사건 범행의 도구로 사용한 부엌칼은 날 길이가 19cm에 달하고 마침 이 사건 범행 당일 피고인이 숯 돌로 갈아 두었던 것으로 날이 날 카로 워 사람을 죽이거나 치명상을 가할 수 있는 위험한 물건에 해당하는 점, ② 피고인은 위 부엌칼로 피해자의 명치 및 배 부위를 찔러 피해자에게 간과 대망(위 근처에 있는 기름조직)에 깊은 자상을 가하였는바, 간 부위에 대량 출혈이 발생할 위험성으로 인하여 피해자가 사망할 가능성이 상당하였으며, 위와 같이 부엌칼로 사람의 복부를 찌를 경우 생명의 유지에 필수 적인 기관들이 손상되거나 과다 출혈 등으로 사망의 결과에 이를 위험성이 매우 크다는 사실을 피고인으로서는 충분히 예견할 수 있었던 점, ③ 피고인이 부엌칼로 피해자의 명치 및 배 부위를 찌른 후 재차 피해자의 목 부위를 찌르려고 하였으나 피해자가 이를 오른팔로 막아내는 과정에서 피해자의 오른팔에 상처가 생긴 것으로 보이는 점 등 피고인이 범행에 이르게 된 경위, 범행의 동기, 준비된 흉기의 유무· 종류· 용법, 공격의 부위와 반복성, 사망의 결과 발생 가능성 정도 등 범행 전후의 객관적인 사정을 종합하여 보면, 이 사건 범행이 비록 순간적인 충동에 의하여 우발적으로 일어난 것이라고 하더라도 피고인은 자신의 행위로 인하여 피해자가 사망할 가능성 또는 위험이 있음을 인식하거나 예견하였다고
1:할 것이다.

Figure 13. Example Output of Kkma Sentence Detection

While the package can be used to split sentences from the document, it is not suited to detect phrases. Its detection algorithm also split the sentence into unnecessary chunks (e.g., line 1 in Figure 13 can be roughly translated to "will do" and does not contribute to the argument analysis). Therefore, we proceeded to work with the keyword match method.

▪ *Keywords*

In our dataset, punctuation is a good indicator when detecting clauses. However, commas are also used to list items instead of indicating the start of a new phrase, and periods are sometimes used as abbreviation marks or show dates (e.g., 2014.10.06). To circumvent such cases, we check if the part of speech of the word before the punctuation. Korean sentences usually end with a verb; thus, we check if the punctuation is preceded by a verb and split the sentence accordingly.

Keywords (or markers) used to split sentences in our dataset were:

Table 4. Keyword Markers for Sentence Splitting

Punctuation	comma, period, quotation marks
Words	“점”, “등”, “로”

Research on carving meaningful phrases out of sentences is a natural language processing task beyond this study's scope. Thus, we used keywords to separate sentences into phrases and annotated the phrases as argument units. We manually edited sentences that were not adequately split before the annotation process.

For convenience and uniformity, we call phrases in our dataset sentences.

3) Annotation using Argumentation Schemes

a. Argument annotation scheme

Previous research in argument mining uses a simple premise-conclusion categorization for the argument components [22], [34], [42], [75]. However, simply annotating premises and conclusions do not show how the components interact

to build an argument. Another issue we have encountered while attempting to annotate using the premise-conclusion structure was categorizing evidence or testimonies and how to handle legislation that supports a premise or conclusion. This coincides with Toulmin's idea that argument analysis involves more than two elements [76, p. 219].

To overcome the limited interpretation of arguments, we have decided to use an adaptation of Toulmin's argumentation model to annotate our data as it identifies the characteristics and role of a statement in the argument and is well-suited to be applied to legal documents [77].

In our adaptation, we have removed the qualifier. The qualifier in Toulmin's model is a modal operator included in a sentence [78, p. 189], and added another component we named *rebuttal-support,* which are statements that support the rebuttal. The table below shows the description of the argument components we gave to our annotators.

Table 5. Argument Component Type and Description

Argument Component	Description
Claim (C)	The conclusion and the heart of the issue. Nees to be identified first. *"What do you want to claim?"* *"What is the argument you are trying to convince?"*
Datum (D)	Grounds supporting the claim. *"What is the basis for supporting the claim?"* *"What are the facts that must be premised for the claim?"*
Warrant (W)	A logical bridge between datum and claim. *"What statement is needed to connect the claim to datum"?*
Backing (B)	Acceptability of the warrant. *"Can you safely reach the conclusion with the warrant?"* *"What else is needed to support the warrant's credibility?"*

	Backing includes common knowledge, legislation, and precedent cases. A statement that explains the application of such information to the case is regarded as a warrant.
Rebuttal (R)	A statement against the claim. *"What must be true for this claim to be false?"* *A rebuttal must have a claim that it tries to defeat.*
Rebuttal-Support (RS)	Grounds supporting the rebuttal. *"What are the facts that support the rebuttal?"*

A short procedure plan was given; a Top-down approach was recommended:

(1) Identify the main claim
(2) Classify the sentence as argumentative or non-argumentative text
(3) Categorize the sentence by the speaker (defendant, prosecution, judge)
(4) Analyze whether the sentence is factual or an assertion of an idea or statement
(5) Which role does the sentence play in Toulmin's argumentation model?
e.g., A factual statement by the judge that supports a warrant is a backing

The application of the adapted Toulmin model on our court decision data can be represented in Figure 14.

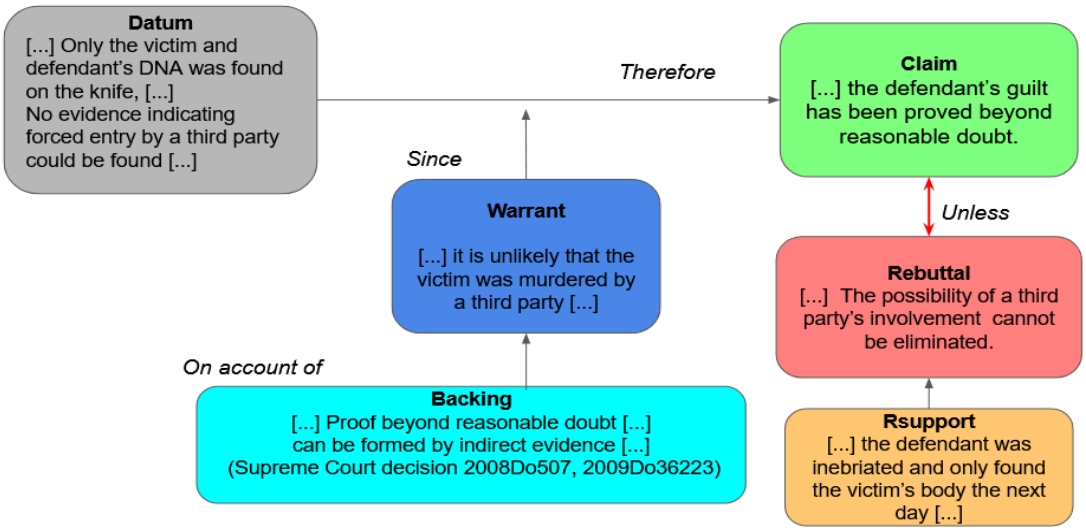


Figure 14. Example Application of Adapted Toulmin Model (2018고합276)

b. Available annotation Tools

There are several annotation tools available to generate text data for natural language processing.

Table 6. Overview of Available Annotation Tools

Name	Description	Source
brat	*Can display the relationship between words *Name entity recognition provided *Annotation tool for sentence analysis	https://brat.nlplab.org/
doccano	* Name entity recognition provided *Text annotation tool for text classification	https://doccano.herokuapp.com/
INCEpTION	* Text annotation available on text and pdf files *Provides Inter-rater reliability calculation *Data extraction possible in various NLP formats	https://inception-project.github.io/
ATLAS.ti	* Various file annotations such as text, pdf file, video, audio, etc. *Provide annotation matching calculation (Krippendorff cu-alpha family) * Convenient project management system	https://atlasti.com/

Except for ATLAS.ti, all annotation tools on the list are free to use. While these tools are useful, they also focus on annotating the sentence's characteristics, especially brat analyzes the relationship between words in the given data using Named Entity Recognition.

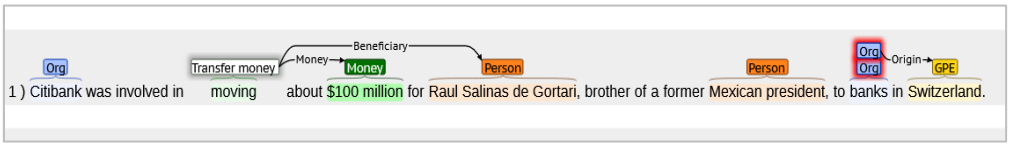


Figure 15. Example brat annotation tool[9]

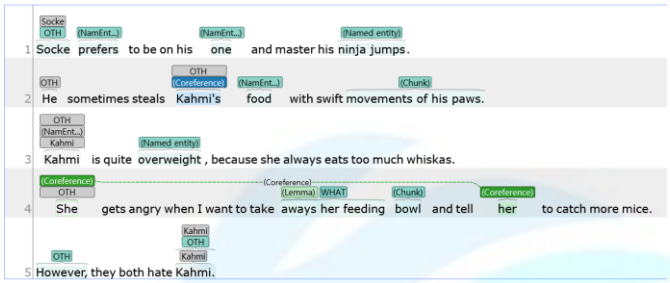


Figure 16. Example INCEpTION[10]

INCEpTION is more suited to analyzing multiple documents. It provides project folders that can be shared with other users; it is also not designed to analyze the argumentative relationships between sentences within the document focusing more on individual words or a span of words.

[9] https://brat.nlplab.org/; screenshot from demo file: tutorials/news/000-introduction.

[10] https://inception-project.github.io/; screenshot from demo file: Concept Linking/ pets2.txt.

c. *Developing an Annotation Tool*

To reduce the error rate when computing the annotated data and help annotators focus only on the annotation process itself, we have developed a simple annotation tool. The tool (named "CaseMark", See Figure 17) loads text data into numbered cells, which can be tagged with a simple mouseclick. It is built on electron (9.2.0) [11], an open-source software framework using HTML, CSS, and javascript to provide cross-platform support. Electron forge (6.0.0-beta.52) was used to manage and develop the electron application.

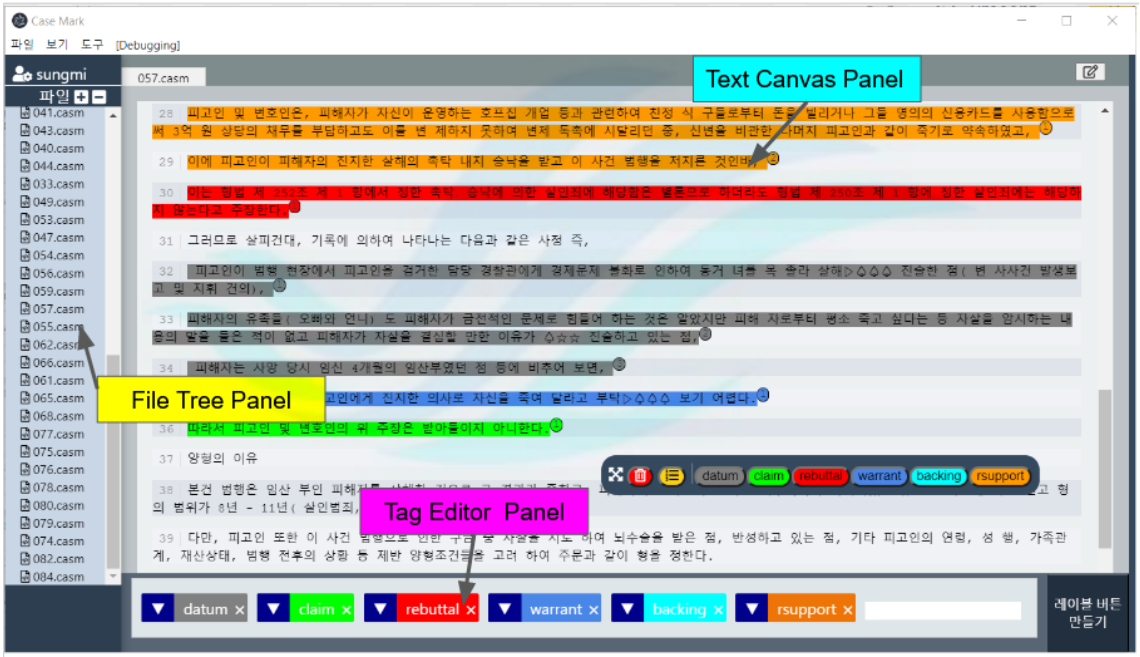


Figure 17. CaseMark Tool Interface

The goal of CaseMark is to provide a simplistic and intuitive annotation tool in which settings can be shared easily with other users (i.e., coders or annotators). The tagged file can be saved as a ".casm" file, a JSON file containing metadata

[11] https://www.electronjs.org/

such as filename, tag settings, coder name, and content data such as the individual line and its matching tag. This enables the user to read the annotated file of others even if they do not share the same tag settings. An overview of the functions is shown in Table 7.

Table 7. Function Description of the CaseMark tool

Panel	Function	Description
File Tree Panel	Add file	Add file to file tree panel. The file can be selected and loaded to text canvas with a double click. Currently, supported file formats are plain text, html, and casm.
	Remove file	Remove file from file tree panel.
	Add folder	Adds all supported files in a folder to the panel.
	Edit coder name	Edit the name of the coder. Default is "coder1".
Tag Editor Panel	Add new tag	Add a new tag by writing the name of the tag into the blank field and pressing ENTER.
	Change tag color	A color picker window opens when clicking on the ▼ button on the left side. Select a color using the left mouse click.
	Create a tag button bar	Click on the square button on the right side of the tag editor panel. A movable button bar will appear on the text canvas panel.
Text Canvas Panel	Split data into lines	The text canvas reads the selected file line by line and splits them into individual elements (tag). Each line is numbered on the left side of the text. Editing the content of the line is disabled.
	Add a tag to line	With the tag button bar, each line is taggable. Select the target line with a left mouse click and click on the tag name on the tag button bar. The tagged line will be highlighted in the color of the selected tag.
	Remove tag	A simple right mouse click on the target line will remove the tag from the line.
	Edit lines	The edit button on the right top corner of the text canvas panel disables the tag button bar and focuses on the text canvas. Users can edit the content of the

		line and create a new line element using ENTER.
Tag button bar	Number tags	Add the number of each tag by clicking on the yellow tag numbering icon on the tag button bar. The tags are numbered (and counted) by the group.
	Remove all tags	Remove all tags in the document by clicking the red bin button on the tag button bar.

After the tagging process is completed, the user can export the data into CSV (comma-separated values) files. Lines that contain commas are framed with quotation marks to prevent splitting. These files are the input data that can be used to train the machine learning models. An example of the tagged output is shown below.

```
sungmi,na,형법 제 48조 제 1 항 제 1호
sungmi,na,피고인 및 변호인의 주장에 대한 판단
sungmi,rsupport_1,피고인 및 변호인은 ' 피해자를 살해한 기억이 없다.
sungmi,rsupport_2,이 사건 당시 술에 만취하여 잠이 들었다 깨고 나서야 피해자가 칼에 찔려 사망한 사실을 발견하였다.
sungmi,rebuttal_1,제 3자가 피해자를 살해했을 가능성을 배제할 수 없다.
sungmi,rsupport_3,"이 사건 당시 술에 취하여 심신 상실 또는 심신 미약 상태에 있었다' 고 주장하다가, 마지막 공판 기
일에 이르러서 야 이를 번의하여 자백하였다."
sungmi,na,그러나 여전히 피고인의 자백 취지는 ' 술에 취해 기억은 안 나지만 증거관계상 본인의 범행으로 인정하겠다' 는
것이다.
sungmi,na,이 사건 공소사실을 유죄로 본 사정에 대한 설명이 필요 하다고 판단된다.
sungmi,na,1. 관련 법리
sungmi,backing_1,"형사재판에 있어 유죄의 인정은 법관으로 하여금 합리적인 의심을 할 여지가 없을 정도로 공소사실이 진
실한 것이라는 확신을 가지게 할 수 있는 증명력을 가진 증거에 의하여야 하고, 이러한 정도의 심증을 형성하는 증거가 없다면
```

*Figure 18. Example *.csv Output of an Annotated Document*

3. Evaluation of the Annotated Corpus

1) Inter-rater Reliability

Reliable data should be able to reproduce the results, and it should be verified whether or not the annotators agreed with the analysis criteria [79].

An inter-rater reliability (IRR) evaluation is necessary to determine if the given description of the argument components were adequate to serve as criteria. According to [80], a high IRR value does not necessarily improve the accuracy rate of text classification using machine learning.

However, IRR is useful to confirm whether the annotators' perceptions of the label categorization are in consensus. As we work with multiple argument components, a discrepancy between annotators can substantially affect the result.

Several studies in argument mining conducted IRR evaluation on argument annotated datasets:

Table 8. Inter-rater Agreement Results for Argument Mining in Literature

Title	Year	Test data	IRR evaluation method	Result
Study on the structure of argumentation in case law	2008	ECHR cases; 10 docs, 47 docs(with guideline)	Cohen's kappa (sentence)	0.58(10 docs), 0.75 (47 docs)
Annotating Argument Components and Relations in Persuasive Essays	2014	essays; 90 docs	Percentage(sentence), Fleiss' multi-kappa(sentence), Krippendorff's alpha(sentence), unitized alpha(text);	0.86(percentage), 0.70(Fleiss-kappa), 0.71(a), 0.75(a_u)

On the Role of Discourse Markers for Discriminating Claims and Premises in Argumentative Discourse	2015	news; 88 docs	Cohen's kappa (token, sentence), Krippendorff's unitized alpha	44.2(kappa-token), 45.2(kappa-sentence), 40.2(a_u);
Annotating Argument Schemes	2020	US2016G1tv corpus; 505 inference relations	Cohen's kappa, CASS kappa	0.61 (kappa), 0.75 (CASS kappa)

In most cases, the IRR value was successfully increased by refining the guideline [22], [81]. Several methods can be used to calculate IRR. The table below details the IRR evaluation methods used in the literature regarding text data.

Table 9. List of IRR Evaluation Methods

IRR evaluation method	Description
Percentage	Simplest method. (Number of evaluated sentences in each category / Total number of evaluated sentences) * 100
Cohen's Kappa	One of the most commonly used formulas, but the number of evaluators is limited to two and can only be used for nominal data. For more than two annotators, Fleiss's kappa is used.
u-Alpha [79]	A proposed formula to calculate the IRR from continuum data such as text and video, based on a = 1-(Do/Dc). Not limited to the number of evaluators and data types, and calculates the degree of concordance (reliability) using the entire data (evaluated data, the interval between evaluated data).

2) Result

For this dataset, three legal informatics graduate students were tasked with the annotation. Cohen's kappa is limited to two coders, and for our research, we did not use a continuous text document but separated phrases like sentences, which makes u-Alpha unnecessary. Therefore, we are using Fleiss's variation of kappa and Krippendorff's alpha for inter-rater reliability evaluation.

The result of Fleiss's kappa was *0.7524*, and Krippendorff's alpha was *0.7522*.

Generally, scores above 0.7 are regarded as a good agreement [69], indicating that the annotators were in a consensus of the labels' meaning. The basic statistics of the corpus are shown below.

Table 10. Statistics of Annotated Corpus

Total number of documents (court decisions)	73
Total number of sentences (phrases)	7451
Total number of sentences in the debate section	1876
Total number of arguments	1630

The plot below shows the total number of each label in the corpus. The number of the labels indicates a slight imbalance in data; the count of the datum label is approximately 7.5 times larger than the smallest class – rsupport. An unbalanced dataset is known to cause problems when using machine learning algorithms [82]; however, this dataset's imbalance is not severe [83, p. 19]. Therefore, we will proceed with the research with this dataset.

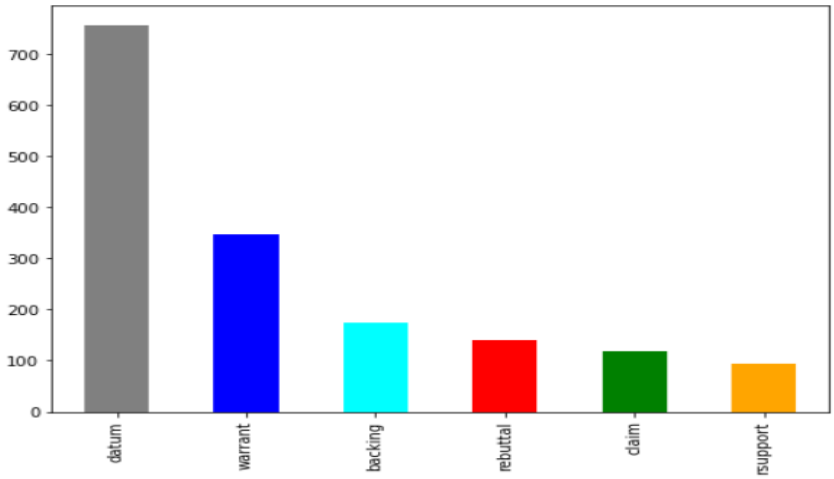


Figure 19. Total Counts of Each Label in Dataset

IV. Research Design

1. Proposed Architecture

In this section, we provide an overview of the proposed alternative hypothesis retrieval model. The architecture of the proposed model has multiple steps (See Figure 20).

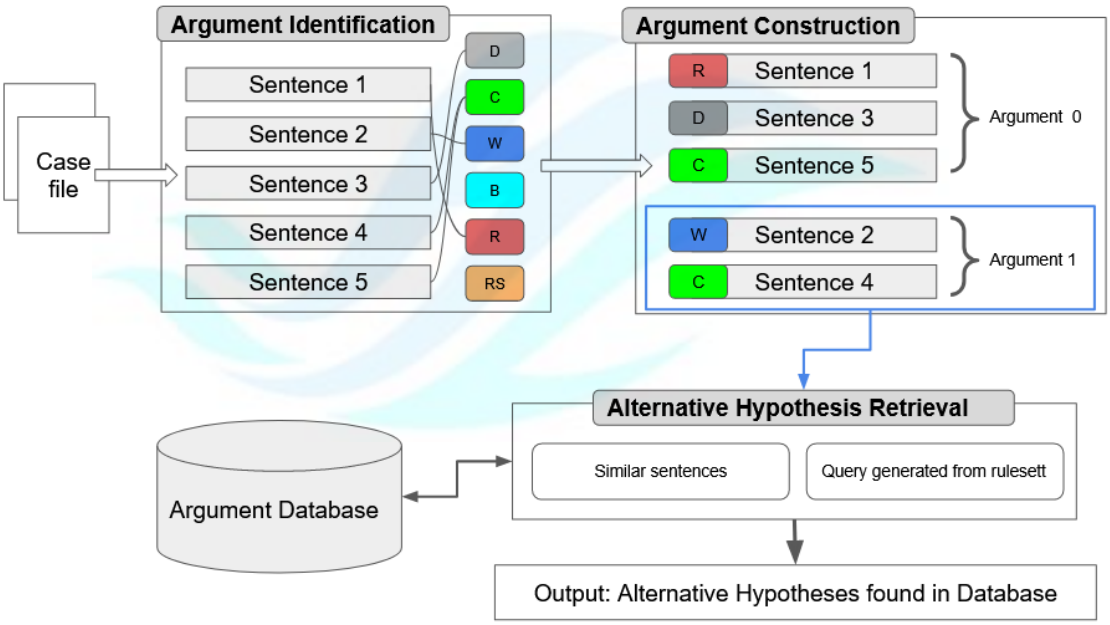


Figure 20. Overview of the Proposed Alternative Hypothesis Retrieval Model

When new cases are loaded, and each sentence is separated as an element. The argument identifier assigns the corresponding label following the adjusted Toulmin argumentation scheme. Then, the labeled sentences are grouped using clustering methods.

An argument group is selected as the query argument chunk. In Figure 20,

argument group 1 is chosen as the query argument chunk. Sentence 2 serves as the left query node, while sentence 4 is the right query node. The data in both sentences are used as input in the alternative hypothesis retrieval system, which computes the cosine similarities between the nodes and sorts the final output according to the computed relevance.

The output is expected to be an argumentative sentence useful to provide a different perspective to the original query. It can be statements that refute the initial claim and facts that can help assess the credibility of the query claim.

The following sections describe the procedure and algorithms used in each process in detail.

2. Argument Component Identification

1) Procedure to Identify Argument Components

Based on previous studies [22], [26], [42], [51], we take a supervised machine learning approach to our annotated data. This part of the study aims to identify the optimal machine learning model with appropriate features that can detect argument components and classify the type. The argument component detection study will be executed in 2 steps.

① Feature extraction: analyze the feature types used in literature and other domain-specific features that can be useful, then adapt them to the data.
② Classification algorithms: testing out the data with the extracted features using several classification algorithms with different parameters.

2) List of Extracted Features

The features extracted for this research follow feature suggestions in the literature. A list of all the features used is given in the table below.

Table 11. List of Features Used in the Analysis

Feature	Description
Unigram	Each word is regarded as a token.
Bigram	Each pair of words are considered a token.
Trigram	Every three successive words are considered a token.
Nouns	Detected using a part-of-speech POS tagger.
Verbs	Detected using a POS tagger.
POS tags	For potential grammatical pattern detection, we have also included POS tags as a feature. [84]
Sentence length	Number of words in a sentence – the word is detected by the POS tagger.
Punctuation	Punctuation marks such as commas, periods, quotation marks are parsed from a sentence.
Section	For this corpus, "Evaluation of Defense" and "Ground of Punishment" can be encoded as features.
Position	The absolute position of the sentence in comparison to the entire document, the calculated values are transformed into [top (~20%), top-mid (21%-40%), middle(41%-60%), middle-bot(61%-80%), bottom(80%~)] [42].
Type of Subject	The sentence's subject is identified through the POS tagger and matched to the manually drafted list of relevant parties. For this study, we will use two options based on the role in

	the argument structure – the “Defendant” and “Others”. [22]
Key Words	Frequency of matching keywords or regex pattern in sentence [47]. The keywords are words that are manually identified as related to arguments. E.g. “그러나 (however)”, “따라서 (therefore)”, “이와 같은 이유로 (Due to this reason)”, “하더라도 (even though)”

One-character length words will be omitted as they usually do not contribute meaning to the sentence and avoid false POS tagging. The *Komoran* class was used for POS tagging. *Komoran* is a relatively new morphological analyzer and can differentiate 42 tags [85]. While *Kkma* can identify more tags, it also the lowest time efficiency.

The *Okt* tagger is one of the fastest (*Mecab* shows the best result but can only be used in a Linux environment), but *Okt* only identifies 19 tags. For this study, *Komoran* is a good compromise.

The type of subject is identified by extracting noun phrases and comparing the first single noun to the predefined list of roles. An example of the parsed tree is shown below. Under the sentence (S), multiple noun phrases are identified; the most left word on the tree (NNP, Proper noun) is “defendant (피고인)”. The type of subject for this sentence will be set accordingly.

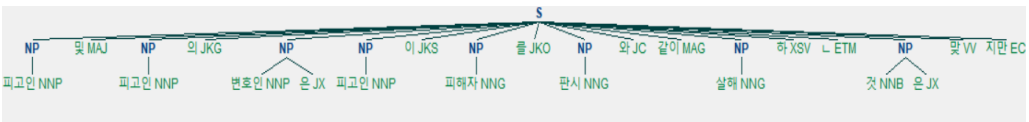


Figure 21. Example Parse Tree for Noun Phrases

Categorical data such as sections and type of subjects were encoded using dummy variables.

An additional column was created with binary feature – 1 for all argument components and 0 for data labeled as "na". The table below shows the total number of features.

The n-gram and POS features were vectorized with Term Frequency-Inverse Document Frequency (TF-IDF) measures.

3) Selected Classification Algorithms

For this research, four classifiers were used: Multinomial Naïve Bayes, Logistic Regression, Support Vector Machine, and Ensemble.

An ensemble classifier is a method that uses multiple classifiers together and finds the class base on a voting rule. Hard voting is a simple majority rule; the most predicted class will be selected as the predicted result. On the other hand, soft voting takes the average result of the predicted probability of each classifier and chooses the class with the largest value as the predicted result. For soft voting, weights can be given to each classifier, which is multiplied with the predicted probability of the respective classifier before computing the average.

If the voting results in a tie, the classifier selects the class in ascending sort order of the label.

In this research, The ensemble classifier was set to hard voting as linear SVM does not provide probabilistic estimations necessary to calculate the average score in soft voting. Parameters that maximize the scores were selected for each classifier.

3. Argument Clustering

1) Procedure to Cluster Arguments

Argument clustering is used to group relevant sentences. Legal documents such as court decisions usually consist of one or more argument groups closely listed together and are referenced by several sentences on a different section of the document. The goal is to use the similarity between the sentences' terms to cluster them to represent the arguments (conclusions).

The procedure we took for this part of the study is stated below:

① Cluster number selection: choose the number of appropriate clusters.
② Cluster the argumentative data in each document: use K-means and Fuzzy c-means clustering on the annotated data and analyze the result.

2) Cluster Number Selection

Determining the number of appropriate clusters does not have one final solution [55]. Dolnicar [55] analyzed two commonly used methods: repeating the calculation with a different number of clusters and use cluster relevant criteria to evaluate the result or use heuristic selections based on corporate criteria. In this study, we use a combination of two methods to iterate the clustering method multiple times and select the number of clusters.

a. Rule-based Approach

The first method that will set the range of cluster numbers to be tested is rule-based. As our data already know each sentence's role in an argument, the potential argument group number should be similar to the number of claims. It

also should not exceed the number of the argument components in each document. It is unlikely that each phrase is its own cluster; therefore, we set the maximum cluster number to the sum of claims and warrants.

Therefore, we set the following rules to determine the range of the number of clusters:

① Set the minimum cluster number to the number of claims.
② Set the maximum cluster number to the sum of (1) and the number of warrants. If the number of warrants is 0, add +1 to (1).

b. Sihouette Coefficient

The silhouette method is a commonly used method to evaluate the data clusters' consistency if the ground truth is not given. The silhouette coefficient for a single sample s is computed as:

$$s_i = \frac{b(i) - a(i)}{max\big(a(i), b(i)\big)}$$

where a(i) is the mean dissimilarity (distance) between object i and every other point in the same class, and b(i) is the mean dissimilarity (distance) between object i and all objects in the next nearest cluster [86]. The calculated result explains whether the model was successful in creating well-defined clusters. A higher silhouette score implies that the objects are a good match for their cluster [86].

We repeat the clustering method with the range of the cluster number set by the rule for our study. Then we use the silhouette coefficient to find the most well-defined cluster number.

3) Selected Features and Clustering Algorithms

We used Word2Vec and n-gram as our main features, as stand-alone or combined with sentence-closeness. For Word2Vec, a Skip-gram model with a context window size of 2 was used. A range of 1 to 3 words was used to generate n-gram features normalized using TF-IDF. Sentence closeness was calculated using the line number of each sentence. A combined approach utilizing all three features was tested as well.

For document clustering, we have utilized K-means and Fuzzy c-means and compared their performance for each document.

4. Alternative Hypothesis Retrieval Model

1) Procedure to Retrieve Alternative Hypotheses

We use a similarity measurement and rule-based approach to find the best alternative hypotheses to a query argument.

We assume that the query is in the form of an argument chunk. By using one sentence and the linked sentences in the argument structure, we believe that it is possible to find more relevant arguments that can help build an alternative hypothesis. Using several similarity values between nodes could also support retrieving more relevant argument components. The figure below visualizes the retrieval procedure based on argument similarity and ruleset.

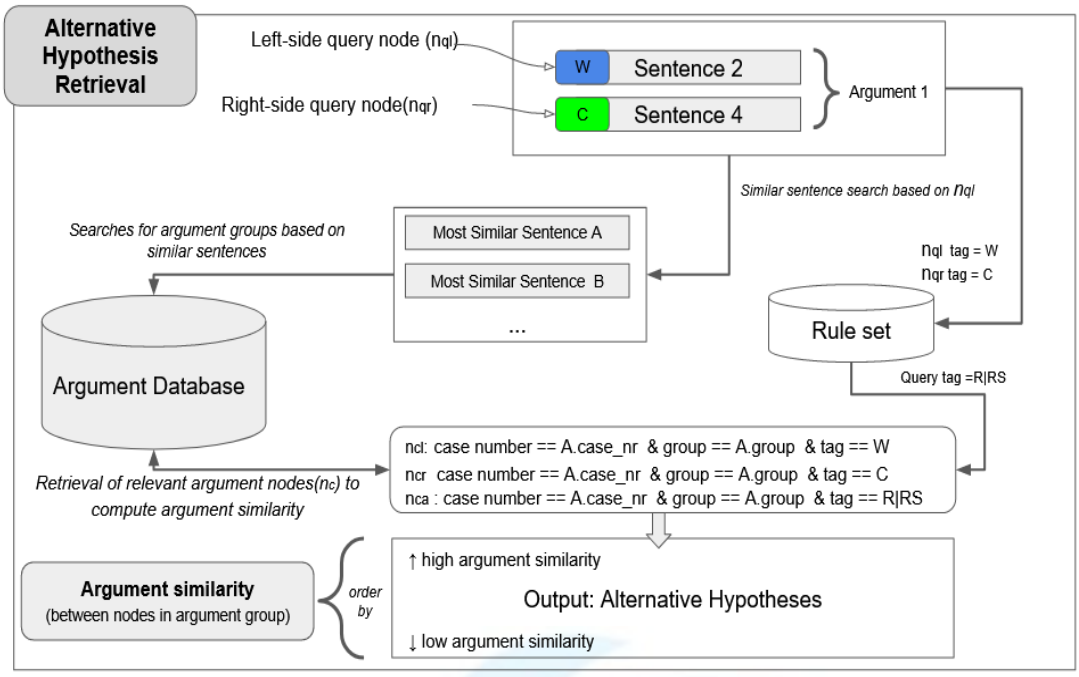


Figure 22. Proposed Alternative Hypothesis Retrieval Procedure

The left-side query node (n_{ql}) is initially used to find the most similar sentences using the Doc2Vec model. The left-side node is chosen as the baseline similarity search, as we assume that the left-side query node contains most of the context, while the right-side node (n_{qr}) is the evaluation or claim of the left-side content.

After the list of most similar sentences is created, the argument structures (case argument group) of each sentence are retrieved from the database. The tag of the query left-side node ("W") and right-side node ("C") is used in the ruleset that determines the component type of the potential alternative hypothesis ("R" or "RS", See 3.b. for details). We use the mean value of several similarities between the nodes (argument similarity between the nodes n_{cl},n_{cr},n_{ca}) to sort the result by relevance.

The summarized procedure of this process is provided below. Note that the primary focus is on "defeating" the target argument; however, this model can be easily used to find supportive arguments as well.

① Find the ten most similar sentences to the left-side query sentence.

② Select the tag type for the alternative hypothesis based on the ruleset.

③ Retrieve the argument groups that contain the sentences identified in ① and calculate the similarities between the nodes (argument similarity).

④ Sort the sentences with the matching tags from ③ by the mean argument similarity value (high to low).

⑤ Show the output of ④.

2) Similarity Measurements for Arguments

We have used a set of different combinations of similarities between the sentence nodes to test our model. First, Doc2Vec is utilized to vectorize the text and compute the similarities between sentences. As we focus on recognizing the relationship between words and sentences and actively aim to understand the role of each component that builds an argument, we also implemented graph similarity as a method to compute the similarity between arguments.

However, we do not strictly follow the computation process proposed by [62]. Finding similar sentences on the same level of the argument structure, i.e., datum to datum, claim to claim, would be ideal, but we have decided against restricting the search process due to the small data size. We also focus on a chunk of the argument structure instead of computing all elements. The search for certain nodes is also guided by a set of rules explained in the next section.

The similarity measurements used in the research are as follows:

sim_g = similarity between the most similar sentence and the alternative sentence

sim_n = similarity between the query left sentence and the alternative sentence

sim_e = edge similarity between the left-side nodes and right-side nodes

$= 0.5 \cdot ((sim_{el}) + (sim_{er}))$

The visual representation of each similarity value between the nodes is shown below.

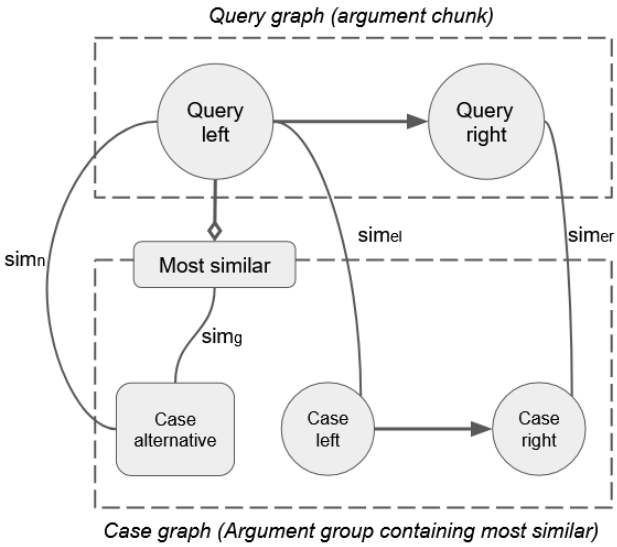


Figure 23. Similarity Measurement of Argument Nodes

The alternative sentence is found based on the alternative argument search rules (See section 3.b). The case-left and case-right nodes share the same tag as their counterparts in the query argument chunk. If more than one node fits the criteria, the node with the highest similarity score to the corresponding query node is selected. The most similar sentence refers to the individual sentences in the top 10 most similar sentence list. Therefore, the model repeats the similarity calculation for each new argument graph retrieved based on the most similar sentence list.

The sentence nodes in the case argument graph can be the same sentence, except case-left and case-right nodes. If the appropriate case-left and right nodes are not found in the case graph, the sim_e value is not included in the overall similarity equation.

3) Rule-based Argument Search

While several similarity measurements can be used to find similar sentences, our goal is not to simply find semantically similar words or paragraphs but to retrieve certain argument components that can help build alternative hypotheses.

For this purpose, we set up rules to use the labels and argument similarity as a query that can retrieve our intended arguments.

In this study, we assume that the investigator requires an alternative hypothesis to an argument chunk. The argument chunks are linked groups around the core components of Toulmin's argumentation model: Datum-Warrant-Claim. We also assume that the node on the left side contains more contextual terms while the right node contains the conclusion (claim) of the argument chunk. Below is a list of the query rules we have found that returns the most satisfactory results from the database.

Table 12. Query rules for Argument Retrieval

Argument chunk		Final query
Label of query left node	Label of query right node	
rsupport	rebuttal	warrant | backing
warrant	claim	rsupport | rebuttal
backing	warrant	rsupport| rebuttal
datum	warrant	datum|rsupport|rebuttal
datum	claim	datum|rebuttal

As the list suggests, the argument type that serves to find the alternative hypothesis to the original query depends on the original query's role. If the original query node contains a defendant's claim, it is better to search for arguments that were asserted by the judge or prosecution.

In our research, we have found that searching for two components connected in the scheme (e.g., backing is linked to warrant, rsupport is a supportive premise or claim to rebuttal) usually results in retrieving a more comprehensive argument suggestion. We also found claims were not suitable for information retrieval as they usually are the direct counter-arguments to rebuttal and do not offer substantial ground for the claim. The detailed results and analysis will be discussed in Chapter V.

V. Result and Analysis

1. Argument Component Identification

In this section, we discuss the result of the argument component identification process. The experiment was conducted using the scikit-learn library. The data was split into two sets of training and test data; one set containing the feature matrix, the other containing the target values ("tag" column). We applied 10-fold cross-validation on the dataset.

1) Evaluation of the Classifiers

Table 13. Results of Argument Component Classifiers

	MultinomialNB (alpha = 0.01)	Logistic Regression (multinomal, C=1.0)	Support Vector Machine (kernel =linear, C=1.0)	Ensemble (voting=hard, weight = [1,2,1])
F1	0.6860	0.7254	0.7466	0.7301
Precision	0.6946	0.6783	0.7082	0.6670
Recall	0.9087	0.9248	0.9329	0.9329

The f1 score (macro) is calculated as the unweighted average of precision and recall, which calculates each label's metrics and finds their average by the number of true instances for each label. Precision is calculated by dividing the count of true positives by the sum of true positives and false positives. Recall scores are the divided score of true positives and the sum of true positives and false negatives.

This result shows that SVM has the best performance when it comes to classifying the argument components. The ensemble classifier that utilizes all

three classifiers achieved the second-highest f1 score, indicating that the majority of votes were in agreement with the SVM model.

In the next section, we will analyze the misclassified data to understand how to improve the models' performance.

2) Analysis of Misclassified Data

The table below shows the number of each sentence that was misclassified. Most classifiers (except the ensemble classifier) are similarly worse in classifying some components while performing well in others. However, naïve Bayes tended to classify non-argumentative sentences into arguments wrongly, while the other classifiers had problems classifying the datum component.

Table 14. Counts of Misclassified Sentences by Argument Component Type

	The total count in corpus	Naïve Bayes	Logistic Regression	Support Vector Machine	Ensemble
Claim	119	7	9	7	8
Datum	756	6	15	22	14
Warrant	348	4	6	6	7
Backing	175	3	3	3	3
Rebuttal	139	5	2	2	5
Rsupport	93	14	12	10	12
na	5823	28	4	6	4

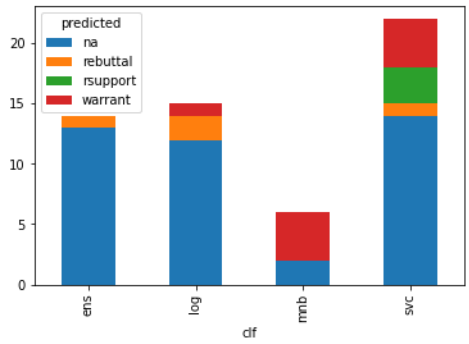


Figure 24. Misclassified Argument Component Plot (actual type - datum)

When analyzing the misclassified datum sentences, it showed that they were mostly classified as non-argumentative sentences. This result could be that non-argumentative sentences and datum sentences both are factual statements, the difference mostly relying on the role the sentence plays in the argument structure.

Naïve Bayes also tended to classify non-argumentative sentences into datum (21 out of 28 were misclassified as datum). This shows misclassification issue of the naïve Bayes classifier also lies in differentiating the factual statements from each other.

Rsupport was the second-highest misclassified sentences. In all classifiers, rsupport sentences were mostly classified as *datum*, as rsupport tends to be the grouds supporting rebuttal clauses. It also has the smallest number in the dataset, which could contribute to the poorer performance than the other classes.

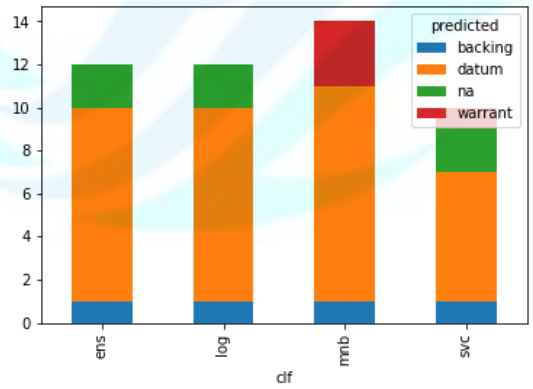


Figure 25. Misclassified Argument Component Plot (rsupport)

The models' result indicates that the Toulmin model can be directly applied to identify each argument component. Adding keyword features and grammatical evaluation for each class (component type) could increase the classes' performance with smaller datasets.

2. Argument Clustering

As we have already analyzed the role of an identified argument component based on the Toulmin scheme, cases that contained only one claim did not need further analysis. Therefore, we filtered cases that had more than two-argument groups and annotated 17 court decisions in total. The number of argument groups in each document is shown in the figure below.

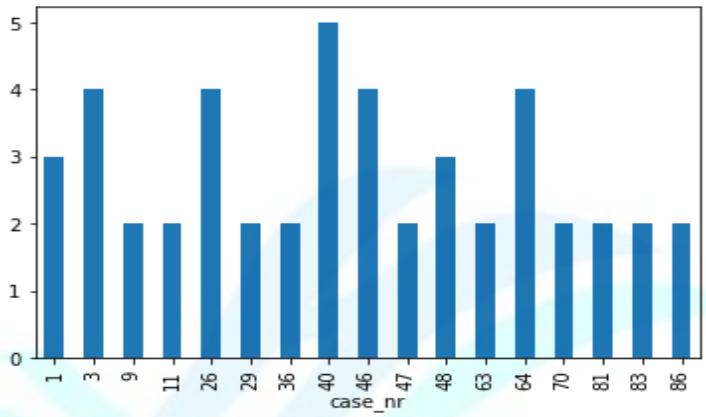


Figure 26. Counts of Manually Analyzed Argument Clusters

The minimum number of groups is 2, while the largest is 5. These groups were based on the relevance between each sentence and structuring the arguments around the claim. Thus, the number of claims is an approximate match of the manually analyzed number of argument groups in the annotated data. We used the annotation as ground truth for the clustering procedure and calculated the f1 (macro) score for each document.

1) Clustering Results

The table below shows the f1 scores for the number of clusters that achieved the highest silhouette scores depending on the features.

We have highlighted the highest results that are above 0.5 in each case.

Table 15. F1 Scores of the Clustering Results

Case	*k*	Word2Vec		Ngram		Word2Vec + Sentence Closeness		Word2Vec+ Ngram		All	
		K-m	FCM	K-m	FCM	K-m	FCM	K-m	FCM	K-m	FCM
1	3	0.30	0.30	0.35	0.12	0.30	0.16	0.53	0.12	0.24	0.34
3	4	0.08	0.19	0.10	0.13	0.23	0.02	0.08	0.17	0.19	0.11
9	2	0.39	0.36	0.33	0.12	0.39	0.36	0.39	0.36	0.39	0.36
11	2	0.36	0.40	0.22	0.36	0.04	0.08	0.35	0.33	0.14	0.07
26	4	0.03	0.05	0.12	0.07	0.17	0.07	0.24	0.27	0.03	0.03
29	2	0.13	0.72	0.17	0.18	0.21	0.76	0.29	0.61	0.21	0.02
36	2	0.48	0.32	0.30	0.24	0.48	0.48	0.48	0.32	0.34	0.27
40	3	0.24	0.32	0.33	0.16	0.24	0.20	0.23	0.40	0.24	0.40
46	3	0.28	0.23	0.21	0.17	0.28	0.28	0.19	0.24	0.19	0.24
47	2	0.53	0.24	0.47	0.37	0.15	0.24	0.53	0.24	0.31	0.24
48	3	0.06	0.39	0.20	0.15	0.06	0.18	0.06	0.19	0.09	0.18
63	2	0.49	0.49	0.15	0.31	0.49	0.32	0.29	0.16	0.29	0.67
64	4	0.11	0.11	0.27	0.21	0.11	0.11	0.11	0.16	0.11	0.10
70	2	0.26	0.26	0.24	0.29	0.26	0.26	0.28	0.26	0.28	0.22
81	2	0.11	0.43	0.04	0.41	0.11	0.35	0.35	0.35	0.35	0.35
83	2	0.42	0.39	0.29	0.27	0.31	0.95	0.46	0.40	0.41	0.48
86	2	0.41	0.37	0.18	0.15	0.41	0.40	0.41	0.27	0.41	0.52

Generally, the larger number of features indicated better performance in both K-means and Fuzzy c-means. Both word2vec and n-gram worked better when combined with sentence closeness.

However, the clustering results are varied, showing good performance on some documents while performing poorly on others. One clear pattern that affects poor clustering performance is the number of clusters. Clusters with higher cluster numbers (outlined cells) show the worst clustering results.

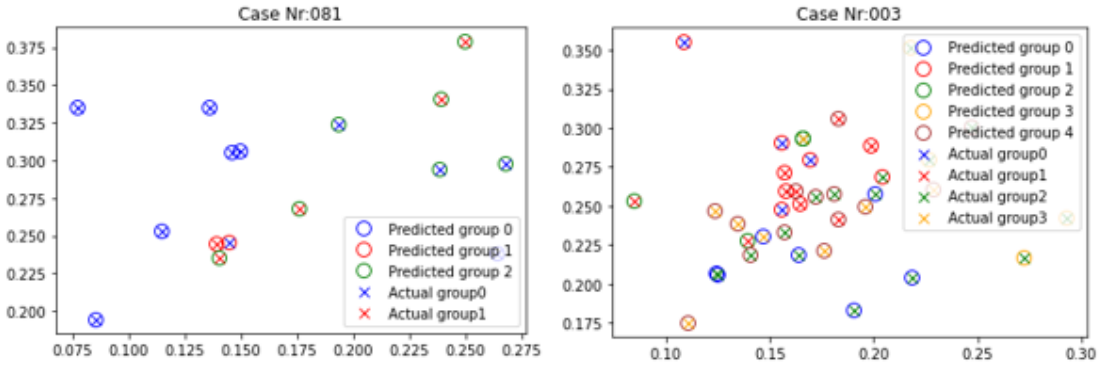


Figure 27. Scatter Plot Visualization of Clustering Results

This can be observed in the plots above. The document (case 081) with the smaller number of clusters matched the prediction to the actual group (21 sentences out of 32 were matched correctly with the gold-standard). In contrast, the cluster with 4 clusters shows almost no matches.

We also compared whether the claim number can be used to solve the cluster number selection problem. The tables below show f1 scores for the cluster numbers in the gold-standard (k_g) and the cluster numbers with the highest silhouette scores.

Table 16. Comparison of F1 Scores Using Different Cluster Numbers

Case	K-means	
	k_g	k_s
1	0.4379	0.3906
3	0.3541	0.3031
11	0.1813	0.2081
26	0.4282	0.0117
40	0.3223	0.3189
46	0.3531	0.2532
48	0.3826	0.1224
64	0.1393	0.0983
81	0.2750	0.6240
86	0.1767	0.4278

Case	FCM	
	k_g	k_s
1	0.5953	0.6116
3	0.2759	0.1479
11	0.1591	0.1134
26	0.2882	0.0117
29	0.6492	0.0370
36	0.6708	0.4020
40	0.4319	0.4319
46	0.3286	0.3050
47	0.3333	0.3915
48	0.3942	0.3712
64	0.2435	0.1522
81	0.2750	0.6240
83	0.8422	0.6369

The scores show that the claim number does not show better performance in both clustering methods than the cluster number obtained using the silhouette method.

2) Example of Argument Clustering and Limitations

An example of the automatically clustered sentences is shown below (all features, FCM clustering).

Table 17. Example of Argument Clustering

Case	Group	Sentence(KOR-original)	Sentence(EN-translated)
63	1	정신질환의 종류와 정도, 범행의 동기, 경위, 수단과 태양, 범행 전후의 피고인의 행동, 반성의 정도 등 여러 사정을 종합하여 법원이 독자적으로 판단할 수 있는 바	Considering the type and degree of mental illness, the motive of the crime, the course of the crime, the means and circumstances, the defendant's behavior before and after the crime, the degree of reflection, etc., this court can decide (…)
63	1	이 법원이 적법하게 채택· 조사한 증거들에 의하여 인정되는 이 사건 범행 경위와 방법, 음주 후 범행 발생 시까지의 시간적 간격	The background and method of the crime in this case and the time interval between the time of the crime after drinking alcohol are accepted based on the evidence selected and investigated by this court (…)
63	1	평소 주량, 범행의 구체적 내용 및 범행 후의 정황, 피고인의 태도 등에 비추어 보면,	Considering the usual alcohol consumption, the specific content of the crime, the circumstances after the crime, and the attitude of the defendant (…)
63	1	피고인이 위 범행 당시 주 취로 인하여 사물을 변별하거나 의사를 결정할 능력이 미약한 상태에 있었다고는 보이지 아니하므로,	At the time of the crime, it does not appear that the defendant was lacking the capacity to discriminate objects or make decisions due to the alcohol consumption (…)

63	1	위 주장도 받아들이지 아니한다.	The defendant's claim is denied.
63	1	위 주장도 받아들이지 아니한다.	The defendant's claim is denied.

This result shows that sentences share similar or related terminology, thus enabling clustering methods that utilize semantic features. However, this relation can be a double-edged sword, as it is shown in the last two rows. While the sentences are identical, they are claims to two different argument groups. As judges tend to used similar sentences to summarize their claims, it becomes harder to separate them.

Another problem is when the sentences are listing different facts to support a common conclusion. In such cases, the sentences do not share terminologies and are mostly clustered into different groups.

This indicates that using clustering methods is not enough to group argumentative sentences automatically. A combined approach using discourse markers [23], [46] and clustering could solve the problem.

3. Alternative Hypothesis Retrieval

We used different sets of similarity values for the analysis to find the most effective method for alternative hypothesis retrieval.

While the original query node is the sentence that initially searches the database for an argument structure containing a similar sentence, it is the similarity between other nodes that determines the ranking order.

Our main assumption for choosing this approach is that the result must share the original sentence's topic and be similar to the other nodes involved, including the linked node to the original query node, which serves as either another premise or conclusion supporting the original node.

The dataset we used for this section of the study is the argumentative text in the original 73 court decisions, in which only the cases with two identified claims

were manually annotated for argument groups.

The rest of the data with only one claim were uniformly assigned to be in one argument group for each document.

1) Alternative Hypothesis Retrieval Result and Analysis

The query left and right nodes were taken and manually identified as an argument chunk from a court decision (2017고합81) not in the dataset.

(KOR)

(1) 이에 피고인이 [피해자의 자해] 를 막기 위해 피해자로부터 칼을 빼앗았으며, 그 후 피해자에게 칼을 빼앗기지 않으려고 단순히 실랑이하는 과정에서 피해자에게 상해가 발생한 것이지, 피고인이 피해자를 칼로 찌르거나 벤 것이 아니고 (rsupport, query-left-node)
(2) 또한 그와 같은 상해 경위에 비추어 보면 당시 피고인에게 살인의 고의가 있었다고 볼 수 없다. (rebuttal, query-right-node)

(ENG)

(1) The defendant took the knife from the victim to prevent [self-harm of the victim], and the victim was injured in the struggle to steal the knife back; the defendant did not stab or cut the victim (rsupport, query-left-node)

(2) also, in the light of how the injuries occurred, the defendant did not act with the intention of murder. (rebuttal, query-right-node)

We have conducted four tests with different sets of similarity measurements (Test 2-5) and one test using the most similar sentence value as the basis (After the relevant argument group was selected based on the most similar sentence list, no other similarity was calculated for Test 1).

Table 18. Retrieved Alternative Hypotheses by Similarity Measurement

Similarity Measurement	Alternative Hypothesis Retrieval for rsupport-rebuttal				
	Test 1	Test 2	Test 3	Test 4	Test 5
sim_g	No	Yes	Yes	Yes	Yes
sim_n	No	No	No	Yes	Yes
sim_e	No	No	Yes	No	Yes
Retrieved alternative hypotheses (Top 5 tags)	458	166	458	156	156
	315	73	393	166	393
	313	156	166	73	73
	329	227	73	57	166
	55	228	156	227	442

Compared to Test 1, the rest of the tests retrieved similar sentences (three out of five are the same) as the most relevant alternative hypotheses. However, matching results alone cannot serve as performance measurement. As there is no evident ground truth, the performance of this model had to be subjectively evaluated. Our primary concern is to find argument statements that can serve as alternative hypotheses. Therefore, we have analyzed the result primarily based on its context.

When evaluating the final output, we found that the results from Test 2,3,4, and 5 were generally more satisfactory than the sentences from Test1.

The sentences in the database corresponding to each tag are shown in the tables below.

Table 19. Overview of Retrieved Sentences - Test 2-5

Tag Nr	Sentences Commonly Retrieved in Test 2-5
156	칼에 찔린 상처의 부위 및 개수, 깊이 등에 비추어 볼 때 이는 피고인의 주장과 같이 넘어지면서 우연히 칼날이 피해자의 가슴에 꽂혀 발생한 자상이라고 보기 어렵고, Considering the area, number and depth of the stabbed wound, it is difficult to believe the defendant's claim that the wound was accidentally inflicted on the victim's during their fall,

73	피고인이 칼로 피해자를 찌를 당시 자신의 행위로 인하여 피해자의 사망이라는 결과가 발생할 가능성 또는 위험이 있다는 것을 충분히 예견할 수 있었음에도 이를 용인하였다고 봄이 상당하므로, It can be acknowledged that the defendant was able to sufficiently predict that there was a possibility or danger of the death of the victim due to his actions when he stabbed the victim and yet proceeded with the act,
166	⑤ 피해자의 왼쪽 가슴 중 위에서 본 자창 주변에 칼끝에 의한 것으로 추정되는 예기손상이 10개 가량 발견되는데,* ⑤ About 10 wounds presumably caused by the tip of a knife can be found on the victims' left chest,* **While this sentence fragment by itself does not assert a claim, it was used to support an inference (warrant) in the original data. In this instance, it complements tag 156.*

Tag 155, 73, and 166 are all shared as the most relevant alternative hypotheses in Test 2, 3, 4, and 5. Tag 155, 166 shows the judge's evaluation regarding stabbing wounds using a knife, 166 explaining the specific wound's circumstance, and 155 reasoning why the defendant's claim is unlikely. Tag 73 states the reason a judge has accepted *dolus eventualis* (willful negligence, 미필적 고의) in a case where the defendant has stabbed the victim. The retrieved sentences could form an alternative hypothesis together or suggest potential directions to form a new hypothesis.

For example, the defendant's claim in the query-left-node ("the injuries on the victim was the result of a struggle to prevent self-harm") could be debated by the fact that the number, area, and depth of the wound is too severe to have happened *without* intent to harm (Tag 155 and 166). Intent to murder, the defendant's claim in query-right node, can also be evaluated based on Tag 73; by stabbing the victim, the defendant's willingness to harm despite the possibility of the victim's death can be acknowledged.

Table 20. Retrieved Sentences - Test 3 and 5

Tag Nr	Sentences Retrieved in Test 3 and 5 (including edge similarity)
393 [Test 3 &5]	살피건대, 형사재판에서 유죄의 인정을 저지하는 합리적 의심이라 함은 모든 의문, 불신을 포함하는 것이 아니라 논리와 경험칙에 의하여 요증사실과 양립할 수 없는 사실의 개연성에 대한 합리적 의문을 의미하는 것으로서, 피고인에게 유리한 정황을 사실 인정과 관련하여 파악한 이성적 추론에 그 근거를 두어야 하는 것이므로, Reasonable doubt in a criminal trial does not include all doubts and distrust, but refers to reasonably questioning about the likeliness of events occuring that are incompatible with the facts using logic and rule of thumb, circumstances in favor of the defendant should be based on rational reasoning,
442 [Test 5]	이 사건 범행이 비록 순간적인 충동에 의하여 우발적으로 일어난 것이라고 하더라도 피고인은 자신의 행위로 인하여 피해자가 사망할 가능성 또는 위험이 있음을 인식하거나 예견하였다고할 것이다. Even if the crime, in this case, occurred accidentally due to a momentary impulse, it can be said that the accused had recognized or predicted that there was a possibility or danger of the victim's death due to his actions.
458 [Test 3]	피고인이 조**, 배** 과 공모하여 이후 피해자 보험회사에 허위로 보상 접수를 한 사실도 인정할 수 있으므로, It can also be admitted that the defendant has made a false claim to the victim's insurance company after conspiring with Joe ** and Bae **.

Our initial assumption regarding edge similarity was that it would give the retrieval process more stability by selecting argument structures that share similar topics and relationships. Tag 393 and 442 are relevant to the topic; however, tag 458 shows no contextual relation to the original argument. Tag 458 is also shared in Test 1 as the first alternative statement. This indicates that one sentence (*the most similar sentence*) in tag 458's argument structure was similar to the original sentence, while the sentence that should have served as the alternative sentence did not.

This shows that the edge similarity by itself should not be the deciding factor to rank alternative hypotheses. This could be explained by the fact that the query tag "rsupport" is the smallest class in the dataset and often cannot be found in the argument group. The edge similarity for this specific query was often omitted, as there was no corresponding "rsupport" tag in the argument group. Without sim_n, edge similarity values could lead to inaccurate results to balance out the discrepancy.

Table 21. Retrieved Sentences – Test 2 and 4

Tag Nr	Sentences Commonly Retrieved in Test 2 and 4 (including node similarity)
227	자기의 행위로 타인의 사망이라는 결과를 발생시킬 만한 가능성 또는 위험이 있음을 인식하거나 예견하면 충분한 것이고 그 인식이나 예견은 확정적인 것은 물론 불확정적인 것이라도 이른바 미필적 고의로 인정된다. It is sufficient to recognize or predict that there is a possibility or risk that the act may result in the death of others, and that recognition or prediction can be definite or undetermined, to be acknowledged as willful negligence.

Test 2 and Test 4 both ranked tag 227 as their fourth and fifth hypothesis. Tag 227 describes willful negligence; more importantly, the criteria willful negligence can be accepted.

Tag Nr	Other Sentences in Test 2 and 4
57	피고인이 누워 있는 피해자를 뒤에서 칼로 찌른 이 사건 범행은 피해자의 피고인에 대한 현재의 부당한 침해를 방위하거나 그러한 침해를 예방하기 위한 행위로 상당한 이유가 있는 경우에 해당한다고 볼 수 없다. This crime, in which the defendant stabbed a lying victim with a knife from behind, cannot be considered as a case for which there is sufficient reason for the defendant to defend against or prevent a currently occurring unjust violation act by the victim.

228	피고인이 범행 당시 살인의 범의는 없었고 단지 상해 또는 폭행의 범의만 있었을 뿐이라고 다투는 경우에 피고인에게 범행 당시 살인의 범의가 있었는지는 피고인이 범행에 이르게 된 경위, 범행의 동기, 준비된 흉기의 유무· 종류· 용법, 공격의 부위와 반복성, 사망의 결과 발생 가능성 정도 등 범행 전후의 객관적인 사정을 종합하여 판단하여야 한다. If the defendant argues that there was no intent to murder at the time of the crime, but only intent to cause injuries or assault, the objective circumstances before and after the crime including the motive, usage, type of the weapon, area, and repetition of attack and the degree of likelihood of death, must be evaluated in a comprehensive manner to determine the defendant's intent,

Tag 57 is the evaluation of the defendant's action (stabbing the victim), while tag 228 is a backing statement for analyzing the intent to murder. The similarity between the alternative hypothesis node and the left query node (sim_n) selects Tag 57 for Test 4 as the sentences share contextual circumstances. Test 2, on the other hand, neglect this value and therefore suggest Tag 228. This shows that the sim_ncan be used to retrieve information depending on the purpose; we can use $sim_n$for more contextual similarities or ignore the value for a more diverse search.

Other results from our experiment are listed in Appendix 4.

2) Example Output of Alternative Hypothesis Retrieval and Limitations

The output below shows the alternative hypothesis retrieval model using all similarity measures:

(KOR)

Query-left (rsupport):
이에 피고인이 이를 막기 위해 피해자로부터 칼을 빼앗았으며, 그 후 피해자에게 칼을 빼앗기지 않으려고 단순히 실랑이하는 과정에서 피해자에게 상해가 발생한 것이지, 피고인이 피해자를 칼로 찌르거나 벤 것이 아니고,

Query-right (rebuttal):
또한 그와 같은 상해 경위에 비추어 보면 당시 피고인에게 살인의 고의가 있었다고 볼 수 없다.

Alternatives (sim_g,sim_n,sim_e):
(tag, sentence)

156:칼에 찔린 상처의 부위 및 개수, 깊이 등에 비추어 볼 때 이는 피고인의 주장과 같이 넘어지면서 우연히 칼날이 피해자의 가슴에 꽂혀 발생한 자상이라고 보기 어렵고,

393: 살피건대, 형사재판에서 유죄의 인정을 저지하는 합리적 의심이라 함은 모든 의문, 불신을 포함하는 것이 아니라 논리와 경험칙에 의하여 요 증사실과 양립할 수 없는 사실의 개연성에 대한 합리적 의문을 의미하는 것으로서, 피고인에게 유리한 정황을 사실 인정과 관련하여 파악한 이성적 추론에 그 근거를 두어야 하는 것이므로,

73:피고인이 칼로 피해자를 찌를 당시 자신의 행위로 인하여 피해자의 사망이라는 결과가 발생할 가능성 또는 위험이 있다는 것을 충분히 예견할 수 있었음에도 이를 용인하였다고 봄이 상당하므로,

166:⑤ 피해자의 왼쪽 가슴 중 위에서 본 자창 주변에 칼끝에 의한 것으로 추정되는 예기 손상이 10 개 가량 발견되는데,

442: 이 사건 범행이 비록 순간적인 충동에 의하여 우발적으로 일어난 것이라고 하더라도 피고인은 자신의 행위로 인하여 피해자가 사망할 가능성 또는 위험이 있음을 인식하거나 예견하였다고할 것이다.

(ENG)

Query-left (rsupport):
The defendant took the knife from the victim to prevent [self-harm of the victim], and the victim was injured in the struggle to steal the knife back; the defendant did not stab or cut the victim,

Query-right (rebuttal):
also, in the light of how the injuries occurred, the defendant did not act with the intention of murder.

Alternatives (sim_g,sim_n,sim_e):
(tag, sentence)

156: Considering the area, number, and depth of the stabbed wound, it is difficult to believe the defendant's claim that the wound was accidentally inflicted on the victim's during their fall,

393: Reasonable doubt in a criminal trial does not include all doubts and distrust but refers to reasonably questioning about the likeliness of events occurring that are incompatible with the facts using logic and rule of thumb; circumstances in favor of the defendant should be based on rational reasoning,

73: It can be acknowledged that the defendant was able to sufficiently predict that there was a possibility or danger of the death of the victim due to his actions when he stabbed the victim and yet proceeded with the act,
166: ⑤ About 10 wounds presumably caused by the tip of a knife can be found on the victims' left chest,

442: Even if the crime, in this case, occurred accidentally due to a momentary impulse, it could be said that the accused had recognized or predicted that there was a possibility or danger of the victim's death due to his actions.

While this retrieval model can find and match alternative hypotheses, the current similarity calculation measurement and the test data show evident limitations. For example, suppose the query sentences do not contain specific enough terms to pinpoint similar arguments. In that case, the result is likely to be a list of generalized arguments against the query type.

Another limitation we have observed is when using warrants and claims as to the query nodes; subjectively, the current similarity measurement is insufficient to find rebuttals and rebuttal-supports relevant enough to be used as alternative hypotheses. This is most likely due to the small dataset for both classes. Experiment results showing the limitation of the model are listed in Appendix 4.

VI. Conclusion

Sense-making for crime investigation analysis is a matter of having domain knowledge and the comprehension of new information, and the application of already collected data. This research aimed to search, test, and propose methods that can support the comprehension and analysis process.

Automatic selection of useful information from natural text and retrieval of related counter-arguments can alleviate an investigator's mental effort; they will be able to focus more on the analysis and evaluation itself than devoting themselves to only the tedious task of filtering documents.

For argument identification, we have confirmed the results of many previous researchers. It is possible to classify sentences into several categories with relatively good performance automatically. However, minority classes were harder to identify. Also, identification becomes difficult if the sentences share a similar context and have to be classified by their role in the argument.

Clustering related argument components prove to be a difficult task. In our experiment, simply using clustering methods with semantic features were not enough to achieve a satisfactory performance; a combined approach utilizing both unsupervised clustering and grouping using discourse markers could bring better results.

Retrieving alternative hypotheses using cosine similarities between sentences and filtering using rules produced a tolerable output. While there are many limitations to the model, including the necessity of a well-classified argument database and lack of diversity in the arguments, we found that the proposed architecture serves as a stepping stone to a better crime investigation evaluation and analysis system.

Overall, this research heavily depends on the initial annotators to analyze the

dataset appropriately so that the majority of the potential users can accept the result. This necessitates the extensive training of annotators and agreement on the argument analysis methods.

References

[1] T. Suh and J. M. Lee, "Save the Lawyer: AI technology accelerates and augments legal work," *IBM Client Sucess Field Notes*, 2018.

[2] W. Song, J. Kim, and E. Chung, "Police want to predict crime with data platform," *Korea JoongAng Daily*, 2017.

[3] P. Pirolli and S. Card, "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis," *Proc. Int. Conf. Intell. Anal.*, vol. 2005, no. January, pp. 2–4, 2005.

[4] F. J. Bex, *Arguments, Stories and Criminal Evidence*, vol. 92, no. 9. Dordrecht: Springer Netherlands, 2011.

[5] S. van den Braak, *Sensemaking software for crime analysis*, no. May. 2010.

[6] B. Verheij, "Automated argument assistance for lawyers," *Proc. Int. Conf. Artif. Intell. Law*, pp. 43–52, 1999.

[7] B. Verheij, "Artificial argument assistants for defeasible argumentation," *Artif. Intell.*, vol. 150, no. 1–2, pp. 291–324, 2003.

[8] N. Fenton and M. Neil, "Decision support software for probabilistic risk assessment using bayesian networks," *IEEE Softw.*, vol. 31, no. 2, pp. 21–26, 2014.

[9] C. Reed and G. Rowe, "Araucaria: Software for Argument Analysis, Diagramming and Representation," *Int. J. Artif. Intell. Tools*, vol. 13, no. 04, pp. 961–979, 2004.

[10] P. Sbarski, T. Van Gelder, K. Marriott, D. Prager, and A. Bulka, "Visualizing argument structure," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5358 LNCS, no. PART 1, pp. 129–138, 2008.

[11] C. Twardy, "Argument maps improve critical thinking," *Teach. Philos.*, vol. 27, no. 2, pp. 95–116, 2004.

[12] F. J. Bex, "Analyzing Stories Using Schemes," *Leg. Evid. ProofStatistics, Stories, Log.*, pp. 93–116, 2009.

[13] N. Pennington and R. Hastie, "Explaining the evidence: Tests of the Story

Model for juror decision making.," *J. Pers. Soc. Psychol.*, vol. 62, no. 2, pp. 189–206, 1992.

[14] R. B. Santos, *Crime Analysis With Crime Mapping*, 4th ed. SAGE Publications, 2017.

[15] C. W. Bruce, S. R. Hick, and J. P. Cooper, *Exploring Crime Analysis: Readings on Essential Skills*. BookSurge, 2004.

[16] S. Gottlieb, S. Arenberg, and R. Singh, *Crime Analysis: From First Report to Final Arrest*. Montclair, CA: Alpha Publishing, 1994.

[17] M. N. Emig *et al.*, *Crime Analysis: A Selected Bibliography*. The Institute, 1980.

[18] IACA (International Association of Crime Analysts), "Definition and Types of Crime Analysis," 2014.

[19] D. M. Russell, M. J. Stefik, P. Pirolli, and S. K. Card, "Cost structure of sensemaking," *Conf. Hum. Factors Comput. Syst. - Proc.*, no. June 2014, pp. 269–276, 1993.

[20] G. Klein, J. K. Phillips, E. L. Rall, and D. Peluso, "A data-frame theory of sensemaking," *Expert. out Context Proc. Sixth Int. Conf. Nat. Decis. Mak.*, pp. 113–155, 2007.

[21] R. J. Heuer, "Psychology of intelligence analysis," *Psychol. Intell. Anal.*, pp. 1–216, 2018.

[22] R. Mochales and M.-F. F. Moens, "Argumentation Mining," *Artif. Intell. Law*, vol. 19, no. 1, pp. 1–22, 2011.

[23] C. Stab, C. Kirschner, J. Eckle-Kohler, and I. Gurevych, "Argumentation mining in persuasive essays and scientific articles from the discourse structure perspective," *CEUR Workshop Proc.*, vol. 1341, no. 1999, 2014.

[24] R. H. Johnson, *Manifest rationality: A pragmatic theory of argument*. 2012.

[25] S. E. Toulmin, *The Uses of Argument*. Cambridge University Press, 2003.

[26] C. Stab and I. Gurevych, "Identifying argumentative discourse structures in persuasive essays," *EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 46–56, 2014.

[27] C. Reed, D. Walton, and F. Macagno, "Argument diagramming in logic, law and artificial intelligence," *Knowl. Eng. Rev.*, vol. 22, no. 1, pp. 87–109, 2007.

[28] C. Reed and G. Rowe, "A pluralist approach to argument diagramming,"

Law, Probab. Risk, vol. 6, no. 1–4, pp. 59–85, 2007.

[29] J. Goodwin, "Wigmore's Chart Method," *Informal Log.*, vol. 20, no. 3, pp. 223–243, 2000.

[30] G. Rowe and C. Reed, "Translating Wigmore Diagrams," *Front. Artif. Intell. Appl.*, vol. 144, pp. 171–182, 2006.

[31] P. Krause, S. Ambler, M. Elvang-Goransson, and J. Fox, "A LOGIC OF ARGUMENTATION FOR REASONING UNDER UNCERTAINTY," *Comput. Intell.*, 1995.

[32] E. Cabrio and S. Villata, "Five years of argument mining: A Data-driven Analysis," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2018-July, pp. 5427–5433, 2018.

[33] I. Habernal and I. Gurevych, "Argumentation mining in user-generated web discourse," *Comput. Linguist.*, 2017.

[34] J. Lawrence and C. Reed, "Combining Argument Mining Techniques," *Proc. 2nd Work. Argumentation Min.*, no. October 2018, pp. 127–136, 2015.

[35] J. Lawrence and C. Reed, "Argument Mining Using Argumentation Scheme Structures," in *Computational models of argument: proceedings of COMMA 2016*, vol. 287, M. S. Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, Ed. IOS Press, 2016, pp. 379–390.

[36] R. Duthie, K. Budzynska, and C. Reed, "Mining Ethos in Political Debate," *Front. Artif. Intell. Appl.*, vol. 287, pp. 299–310, 2016.

[37] M. Lippi and P. Torroni, "Argumentation Mining: State of the Art and Emerging Trends," *ACM Trans. Internet Technol.*, vol. 16, no. 2, 2016.

[38] P. Poudyal, T. Gonçalves, and P. Quaresma, "Using clustering techniques to identify arguments in legal documents," *CEUR Workshop Proc.*, vol. 2385, 2019.

[39] E. Cabrio and S. Villata, "A natural language bipolar argumentation approach to support users in online debate interactions†," *Argument Comput.*, 2013.

[40] S. Teufel, A. Siddharthan, and C. Batchelor, "Towards discipline-independent Argumentative Zoning: Evidence from chemistry and computational linguistics," *EMNLP 2009 - Proc. 2009 Conf. Empir. Methods Nat. Lang. Process. A Meet. SIGDAT, a Spec. Interes. Gr. ACL, Held*

Conjunction with ACL-IJCNLP 2009, no. August, pp. 1493–1502, 2009.

[41] J.-C. Mensonides, S. Harispe, J. Montmain, and V. Thireau, "Automatic Detection and Classification of Argument Components using Multi-task Deep Neural Network Detection and Classification of Argument Components using Multi-task Deep Neural Network Automatic Detection and Classification of Argument Components using," *Proc. 3rd Int. Conf. Nat. Lang. Speech Process.*, no. 1, pp. 25–33, 2019.

[42] T. Goudas, C. Louizos, G. Petasis, and V. Karkaletsis, "Argument Extraction from News, Blogs, and Social Media," in *Artificial Intelligence: Methods and Applications*, vol. 8445 LNCS, 2014, pp. 287–299.

[43] L. E. Allen, "Language, law and logic: plain legal drafting for the electronic age," *Comput. Sci. Law*, pp. 75–100, 1980.

[44] M.-F. Moens, E. Boiy, R. Mochales-Palau, and C. Reed, "Automatic detection of arguments in legal texts," *Proc. Int. Conf. Artif. Intell. Law*, pp. 225–230, 2007.

[45] P. Poudyal, "Automatic Extraction and Structure of Arguments in Legal Documents," University of Évora, 2018.

[46] A. Wyner, R. Mochales-Palau, M.-F. Moens, and D. Milward, "Approaches to text mining arguments from legal cases," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6036 LNAI, pp. 60–79, 2010.

[47] E. Florou, S. Konstantopoulos, A. Koukourikos, and P. Karampiperis, "Argument extraction for supporting public policy formulation," *Proc. 7th Work. Lang. Technol. Cult. Heritage, Soc. Sci. Humanit.*, pp. 49–54, 2013.

[48] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008.

[49] J. Thanaki, *Python Natural Language Processing*. Packt Publishing, 2017.

[50] L. J. Davis and K. P. Offord, "Logistic regression," in *Emerging Issues and Methods in Personality Assessment*, 2013, pp. 273–283.

[51] J. Lawrence and C. Reed, "Mining Argumentative Structure from Natural Language text using Automatically Generated Premise-Conclusion Topic Models," in *Proceedings of the 4th Workshop on Argument Mining*, 2018, pp. 39–48.

[52] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1-12, 2013.

[53] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 1389-1399, Oct. 2013.

[54] P. Lison and A. Kutuzov, "Redefining context windows for word embedding models: An experimental study," *arXiv*, no. May, pp. 284-288, 2017.

[55] S. Dolnicar, "A review of unquestioned standards in using cluster analysis for data-driven market segmentation," *Fac. Commer.*, vol. 273, no. December, pp. 1-9, 2002.

[56] V. Steinbach, M., Karypis, G., Kumar, "A comparison of document clustering techniques," in *KDD Workshop on Text Mining*, 2000.

[57] S. Al-Anazi, H. Almahmoud, and I. Al-Turaiki, "Finding Similar Documents Using Different Clustering Techniques," *Procedia Comput. Sci.*, vol. 82, no. March, pp. 28-34, 2016.

[58] A. Sangalli, *The Importance of Being Fuzzy*. 2018.

[59] L. A. Zadeh, "Is there a need for fuzzy logic?," *Inf. Sci. (Ny).*, 2008.

[60] M. Wright, "Homicide detectives' intuition," *J. Investig. Psychol. Offender Profiling*, 2013.

[61] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," *31st Int. Conf. Mach. Learn. ICML 2014*, vol. 4, pp. 2931-2939, 2014.

[62] M. Lenz, S. Ollinger, P. Sahitaj, and R. Bergmann, "Semantic Textual Similarity Measures for Case-Based Retrieval of Argument Graphs," in *Case-Based Reasoning Research and Development*, 2019, pp. 219-234.

[63] R. Bergmann and Y. Gil, "Similarity assessment and efficient retrieval of semantic workflows," *Inf. Syst.*, vol. 40, pp. 115-127, 2014.

[64] K. D. Ashley, "Applying argument extraction to improve legal information retrieval," *CEUR Workshop Proc.*, vol. 1341, 2014.

[65] R. Rinott, L. Dankin, C. Alzate, M. M. Khapra, E. Aharoni, and N. Slonim, "Show me your evidence - An automatic method for context dependent

evidence detection," *Conf. Proc. - EMNLP 2015 Conf. Empir. Methods Nat. Lang. Process.*, no. September, pp. 440–450, 2015.

[66] D. Walton and T. F. Gordon, *Argument Invention with the Carneades Argumentation System*, vol. 14, no. 2. 2017.

[67] J. Keppens and B. Schafer, "Knowledge based crime scenario modelling," *Expert Syst. Appl.*, vol. 30, no. 2, pp. 203–222, 2006.

[68] A. Chohlas-Wood and E. S. Levine, "A recommendation engine to aid in identifying crime patterns," *Interfaces (Providence).*, vol. 49, no. 2, pp. 154–166, 2019.

[69] C. Stab and I. Gurevych, "Annotating argument components and relations in persuasive essays," *COLING 2014 - 25th Int. Conf. Comput. Linguist. Proc. COLING 2014 Tech. Pap.*, pp. 1501–1510, 2014.

[70] Y. Kim, "Application of Text Mining for Legal Information System: Focusing on Defamation Precedent [KOR]," *J. KOREAN Soc. Libr. Inf. Sci.*, vol. 54, no. 1, pp. 387–409, 2020.

[71] J. Won, J. Jo, and S. Jung, "Extracting information from court case data using Machine Reading Comprehension [KOR]," in *Koran Software Congress*, 2019, pp. 1409–1411.

[72] S. Baek, "Current Status and Future Tasks of the Online Access to Court Records System [판결서 인터넷열람 제도의 개선현황과 향후 과제, KOR]," 이슈와 논점, no. 1571, 2019.

[73] C. Stab and I. Gurevych, "Parsing argumentation structures in persuasive essays," *Comput. Linguist.*, vol. 43, no. 3, pp. 619–659, 2017.

[74] S. Andersen, "Sentence Types and Functions," *San José State Univ. Writ. Cent.*, p. 2, 2014.

[75] M. A. Walker, P. Anand, J. E. F. Tree, R. Abbott, and J. King, "A corpus for research on deliberation and debate," *Proc. 8th Int. Conf. Lang. Resour. Eval. Lr. 2012*, pp. 812–817, 2012.

[76] B. Verheij, "The Toulmin Argument Model in Artificial Intelligence," in *Argumentation in Artificial Intelligence*, G. Simari and I. Rahwan, Eds. Boston, MA: Springer US, 2009, pp. 219–238.

[77] C. C. Marshall, "Representing The Structure of a Legal Argument," in *ICAIL '89: Proceedings of the 2nd international conference on Artificial*

intelligence and law, 1989, pp. 121–127.

[78] D. Hitchcock and B. Verheij, *Arguing on the Toulmin Model: New Essays in Argument Analysis and Evaluation*. Springer Netherlands, 2007.

[79] K. Krippendorff, "Measuring the Reliability of Qualitative Text Analysis Data," *Qual. Quant.*, vol. 38, no. 6, pp. 787–800, 2004.

[80] N. El Dehaibi and E. F. MacDonald, "Investigating Inter-Rater Reliability of Qualitative Text Annotations in Machine Learning Datasets," *Proc. Des. Soc. Des. Conf.*, vol. 1, pp. 21–30, 2020.

[81] J. Visser, J. Lawrence, C. Reed, J. Wagemans, and D. Walton, *Annotating Argument Schemes*. 2020.

[82] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, 2002.

[83] H. He and E. A. Garcia, *Learning from imbalanced data*, vol. 21, no. 9. Springer International Publishing, 2009.

[84] J. Valvoda, O. Ray, and K. Satoh, "Using agreement statements to identify majority opinion in UKHL case law," *Front. Artif. Intell. Appl.*, vol. 313, pp. 141–150, 2018.

[85] KoNLPy, "Korean POS tags comparison chart." [Online]. Available: https://docs.google.com/spreadsheets/d/1OGAjUvalBuX-oZvZ_-9tEfYD2gQe7hTGsgUpiiBSXI8. [Accessed: 21-Dec-2020].

[86] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. C, pp. 53–65, 1987.

[87] K. Cho, "The Unfinished 'Criminal Procedure Revolution' of Post-Democratization South Korea," *Denver J. Int. Law Policy*, vol. 30, no. 3, p. 377, 2002.

[88] J. Choi, "Investigation examiner system, police internal 'complaint' [수사심사관제도, 경찰 내부 '불만', KOR]," 전북일보, 2020.

논증 마이닝 기법을 활용한 범죄 수사 경합가설 탐색에 관한 연구

2020.
석사학위논문
박성미
국제학과
지도교수: 장윤식, 노기영

형사사법시스템이 공판중심주의를 지향하면서 법원, 검찰청, 경찰을 중심을 사실의 재구성, 사건 분석 방법론에 관한 관심이 높아지고 있다. 특히, 2020년 형사소송법이 개정되어 사법경찰의 독립적 수사가 가능케 되면서 경찰 수사관 측 사건 검토 과정이 유례없이 중요해졌다고 볼 수 있다. 그러나 기존의 사건 분석 지원 도구는 논리적인 검증 과정이 아닌 증거 수집 및 분석에 초점을 두고 있는 현황이다. 따라서 효율적인 사건분석과 검토를 위해서는 증거관계와 법적 논리적 쟁점 식별에 도움이 되는 논증 분석 지원 시스템이 요구된다. 본 연구의 목적은 관련 자연어 문서에서 사건분석에 핵심적인 역할을 지닌 (1) 논증 요소를 자동으로 추출하고, (2) 추출된 요소를 그룹화하여 (3) 유사한 경합 가설 (alternative hypothesis) 을 제공할 수 있는 모델 구조를 고안하여 수사관들의 사건 검토 과정을 단축하는 데 있다.

논증 마이닝 (Argument[ation] Mining)이란 단편적인 자연어 정보를 추출하는 기존의 텍스트 마이닝 기법을 넘어 주장과 근거를 식별하고 논증 구조를 분석하는 기술로 정의된다. 본 연구에서는 수사 결과 보고서와 동일한 범죄 사실을 기반으로 사건을 검토하고 평가하는 1심 형사판결문 73건에 변형된 툴민(Toulmin) 모델을 접목하여 분석 후 논증 마이닝 데이터로 활용하였다.

논증 마이닝의 첫 단계인 논증 요소 추출은 관련 선행연구에서 우수한 성적을 보인 Support Vector Machine, Logistic Regression, Naïve Bayes 등 지도학습 문장 분류 기법을 사용하였으며 주로 영문으로 진행되었던 논증 마이닝 기술이 국문 데이터에도 유사 적용될 수 있음을 확인하였다. 또한, 선행연구에서 주로 전제, 결론 또는 대전제, 소전제, 결론의 형식으로 논증 요소를 분류했던 것과 다르게 본 연구에서는 변형된 툴민 논증 모델

을 기반으로 데이터 (datum), 주장(claim), 추론(warrant), 추론의 근거(backing), 대립주장 (rebuttal), 대립주장의 근거 (rebuttal-support) 즉 총 6가지의 논증 문장 분류를 시도했다는 점에서 큰 의의가 있다.

논증 마이닝의 두번째 단계인 논증 구조 재구성은 논증 요소 간의 연관성 분석을 통해 관련 요소를 그룹화할 수 있도록 비지도학습 알고리즘을 활용하였다. Word2Vec, N-gram과 문장 근사성을 피처로 추출하고, 문서 클러스터링에 주로 사용하는 K-평균 (K-means) 군집화 기법과 선행 연구에서 판결문 문장 클러스터링에 우수한 성과 가능성을 보인 퍼지 c-평균 (Fuzzy c-means) 군집화 기법을 사용하여 각 피처와 군집화 기법의 성능을 비교 분석하였다.

마지막으로 본 연구는 논증요소로 분류되고 요소 간의 관계가 정의된 데이터를 통해 경합가설 탐색 모델을 제시한다. 이 모델은 앞선 논증마이닝 기법을 통해 새 문서의 논증 구조가 식별되었다는 가정 하에서 관련 분야의 논증 데이터와의 유사도 측정을 통해 경합가설을 탐색한다. 특히, 단순히 문장과 문장 간의 유사도를 계산하는 것이 아니라 분석된 논증 구조를 바탕으로 경합가설이 될 수 있는 요소 속성을 선택하고 논증 요소 간의 유사도(node similarity)와 논증 구조 간의 유사도(edge similarity) 측정을 포함한다는 차이가 있다. 분석 데이터에 포함되지 않은 판결문의 논증 문장에 본 모델을 적용해본 결과, 1차적인 문장간의 유사도를 통해 경합 가설을 선택하는 것보다 유사 문장을 포함한 논증 구조를 선택하고, 선택된 논증 구조에서 논증 요소간의 유사도를 계산하는 본 모델의 방식이 경합가설로 적합한 논증 요소를 추출하는 데 더 유용하다는 결과를 확인할 수 있었다.

본 연구는 향후 기능 개선을 통해 경합 가설 뿐만 아니라 대립 가설과 근거를 자동으로 제시하고 증거가 부족한 초기 수사 단계에서 수사 방향을 제시할 수 있는 인공지능 수사 시스템의 시금석이 되기를 기대한다.

주제어 : 범죄수사, 논증 마이닝, 자동 논증 요소 추출, 수사 검증, 경합 가설 탐색 모델

Alternative Hypothesis Retrieval Model for Crime Investigation Analysis Using Argument Mining

2020.
Master's Degree
Park, Sung Mi
Department of International Studies
Advisor Prof. Jang, Yun Sik, Prof. Noh, Ghee Young

The Korean National Police became authorized to perform independent investigations due to the revision of the Korean Criminal Procedure Act in 2020. As a result, unprecedented importance was placed on the review process of cases investigated by police. However, existing case analysis support tools do not focus on *logical* verification, tending instead to focus on collecting and analyzing evidence. This fundamental gap in the review and analysis of cases necessitates a support system for *argument analysis*. The purpose of this study is to (1) automatically extract and classify elements of arguments found in related case documents, (2) group these elements, and (3) retrieve potential alternative hypotheses from a repository of these elements.

Argument[ation] Mining is defined as a technology that identifies arguments and evidence and analyzes arguments' structure. To our knowledge, there is no appropriate corpus for argumentation mining available in Korean. We have collected 73 Korean first instance criminal cases, which we analyzed using a modified Toulmin model.

We have selected features based on previous research in argument mining to classify the elements of arguments, especially for the legal domain. However, instead of the usual two- to three types of arguments (premise, claim, the main

claim), we have attempted to classify the sentences into six types of arguments based on the modified Toulmin model (datum, warrant, backing, claim, rebuttal and rebuttal support).

We have used K-means and Fuzzy c-means clustering algorithms to group the argumentative sentences. K-means is a popular clustering method for documents, while a previous research clustering legal arguments proposed fuzzy c-means.

Our alternative hypothesis retrieval model assumes that a new document has been analyzed using the technology stated above. Instead of just finding the most similar sentence to an argument, we use a set of rules to determine the potential alternative hypotheses and use sentence similarity to find a related argument group from the argument repository. Then, we use similarity measurements between the argument nodes and relationships (edge) to retrieve the most relevant alternative hypotheses. Using a new argument from a court decision not included in the initial dataset, we found our model successfully identified relevant alternative hypotheses.

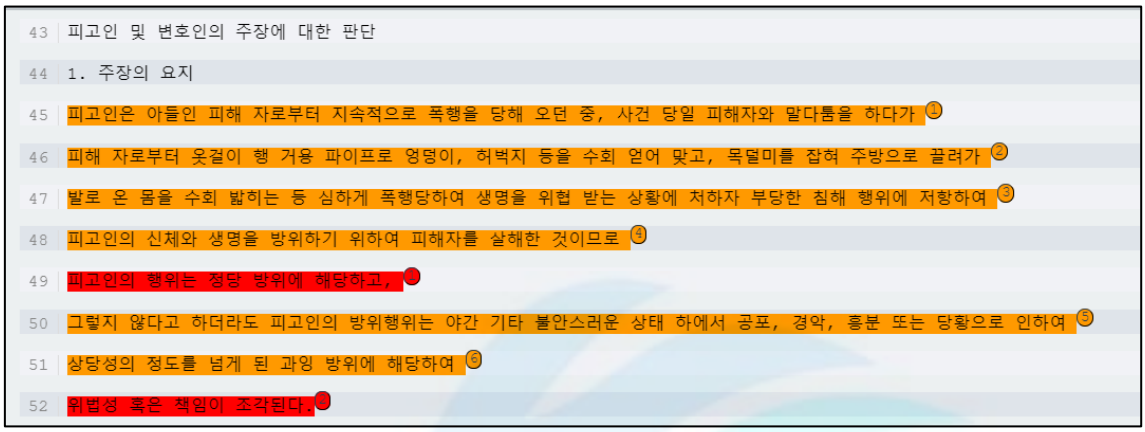
In the future, we hope to develop our model further and enhance the scope and accuracy of the potential hypotheses generation, and ultimately serve as a stepping stone towards developing an Artificial-Intelligence-driven investigation system.

Keywords: Crime Investigation, Argument Mining, Automatic Argument Element Extraction, Investigation Verification, Alternative Hypothesis Retrieval Model

Appendix

<Appendix 1> Example Annotated Court Decision

2011고합207 살인



피고인 및 변호인의 주장에 대한 판단

1. 주장의 요지

피고인은 아들인 피해 자로부터 지속적으로 폭행을 당해 오던 중, 사건 당일 피해자와 말다툼을 하다가 ①

피해 자로부터 옷걸이 행 거용 파이프로 엉덩이, 허벅지 등을 수회 얻어 맞고, 목덜미를 잡혀 주방으로 끌려가 ②

발로 온 몸을 수회 밟히는 등 심하게 폭행당하여 생명을 위협 받는 상황에 처하자 부당한 침해 행위에 저항하여 ③

피고인의 신체와 생명을 방위하기 위하여 피해자를 살해한 것이므로 ④

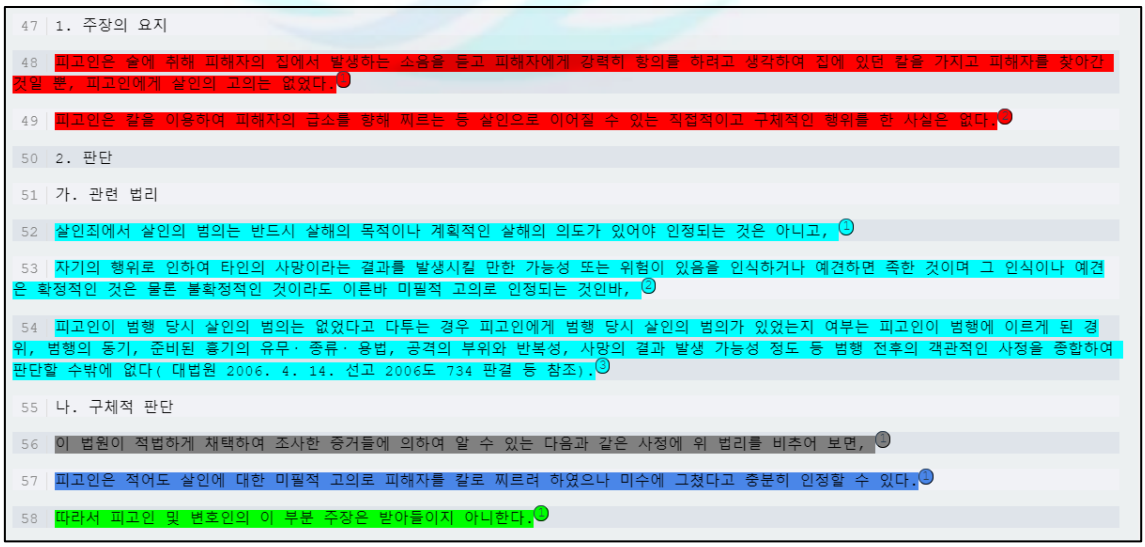
피고인의 행위는 정당 방위에 해당하고, ⑤

그렇지 않다고 하더라도 피고인의 방위행위는 야간 기타 불안스러운 상태 하에서 공포, 경악, 흥분 또는 당황으로 인하여 ⑤

상당성의 정도를 넘게 된 과잉 방위에 해당하여 ⑥

위법성 혹은 책임이 조각된다. ⑦

2019고합267 살인미수



1. 주장의 요지

피고인은 술에 취해 피해자의 집에서 발생하는 소음을 듣고 피해자에게 강력히 항의를 하려고 생각하여 집에 있던 칼을 가지고 피해자를 찾아간 것일 뿐, 피고인에게 살인의 고의는 없었다. ①

피고인은 칼을 이용하여 피해자의 급소를 향해 찌르는 등 살인으로 이어질 수 있는 직접적이고 구체적인 행위를 한 사실은 없다. ②

2. 판단

가. 관련 법리

살인죄에서 살인의 범의는 반드시 살해의 목적이나 계획적인 살해의 의도가 있어야 인정되는 것은 아니고, ①

자기의 행위로 인하여 타인의 사망이라는 결과를 발생시킬 만한 가능성 또는 위험이 있음을 인식하거나 예견하면 족한 것이며 그 인식이나 예견은 확정적인 것은 물론 불확정적인 것이라도 이른바 미필적 고의로 인정되는 것인바, ②

피고인이 범행 당시 살인의 범의는 없었다고 다투는 경우 피고인에게 범행 당시 살인의 범의가 있었는지 여부는 피고인이 범행에 이르게 된 경위, 범행의 동기, 준비된 흉기의 유무· 종류· 용법, 공격의 부위와 반복성, 사망의 결과 발생 가능성 정도 등 범행 전후의 객관적인 사정을 종합하여 판단할 수밖에 없다(대법원 2006. 4. 14. 선고 2006도 734 판결 등 참조). ③

나. 구체적 판단

이 법원이 적법하게 채택하여 조사한 증거들에 의하여 알 수 있는 다음과 같은 사정에 위 법리를 비추어 보면, ①

피고인은 적어도 살인에 대한 미필적 고의로 피해자를 칼로 찌르려 하였으나 미수에 그쳤다고 충분히 인정할 수 있다. ②

따라서 피고인 및 변호인의 이 부분 주장은 받아들이지 아니한다. ③

<Appendix 2> List of Annotated Court Decisions

Nr: case number assigned in the corpus

Arg: count of argumentative text NoArg: count of non-argumentative text

Nr	Arg	NoArg	Court Decision
0	5	56	대구지방법원 2011. 11. 18. 선고 2011고합225 판결
1	41	60	대구지방법원 2011. 9. 9. 선고 2011고합207 판결
3	38	84	대전지방법원 2012. 5. 11. 선고 2012고합31 판결
8	12	52	부산지방법원 2012. 6. 19. 선고 2012고합312 판결
9	11	53	부산지방법원 2013. 5. 10. 선고 2013고합91 판결
10	6	60	부산지방법원 2014. 10. 10. 선고 2014고합557 판결
11	33	82	부산지방법원 2014. 7. 18. 선고 2014고합237, 2014초기1999 판결
12	23	49	부산지방법원 2015. 6. 19. 선고 2015고합37 판결
14	15	46	서울남부지방법원 2012. 11. 22. 선고 2012고합694 판결
15	13	61	서울남부지방법원 2012. 9. 26. 선고 2012고합411 판결
16	14	48	서울남부지방법원 2015. 4. 16. 선고 2014고합570 판결
17	10	62	서울남부지방법원 2016. 7. 21. 선고 2016고합23 판결
18	26	96	서울동부지방법원 2012. 7. 23. 선고 2012고합214 판결
19	7	80	서울동부지방법원 2017. 4. 18. 선고 2017고합6 판결
21	19	134	서울북부지방법원 2017. 4. 27. 선고 2016고합541 판결
23	21	72	서울서부지방법원 2017. 3. 29. 선고 2016고합332 판결
24	10	144	서울중앙지방법원 2010. 10. 8. 선고 2010고합1142 판결
25	46	92	서울중앙지방법원 2012. 1. 17. 선고 2011고합1435 판결
26	47	184	서울중앙지방법원 2014. 3. 28. 선고 2013고합1056 판결
29	26	66	수원지방법원 2012. 2. 24. 선고 2012고합34 판결
30	13	111	수원지방법원 2013. 4. 18. 선고 2012고합1172 판결
31	9	56	수원지방법원 2014. 10. 6. 선고 2014고합441 판결
32	4	66	수원지방법원 2014. 8. 18. 선고 2014고합188 판결
33	11	109	수원지방법원 2014. 9. 11. 선고 2014고합329, 2014초기1827 판결
34	8	61	수원지방법원 2015. 4. 14. 선고 2015고합62 판결
35	12	56	수원지방법원 2015. 4. 7. 선고 2015고합12 판결
36	18	59	수원지방법원 2016. 9. 2. 선고 2016고합309 판결
37	8	74	수원지방법원 2018. 12. 14. 선고 2018고합381 판결
38	264	137	수원지방법원 2018. 5. 18. 선고 2017고합778 판결
39	24	95	수원지방법원 2019. 10. 24. 선고 2019고합267 판결
40	37	60	울산지방법원 2013. 10. 8. 선고 2013고합74 판결
41	15	88	울산지방법원 2013. 11. 19. 선고 2013고합163, 2013감고6 판결
42	11	49	울산지방법원 2013. 5. 24. 선고 2012고합540 판결
43	7	52	울산지방법원 2014. 10. 24. 선고 2014고합179 판결

44	7	71	울산지방법원 2014. 3. 14. 선고 2013고합311 판결
45	32	145	울산지방법원 2015. 2. 3. 선고 2014고합356(분리) 판결
46	38	61	울산지방법원 2015. 5. 8. 선고 2015고합21 판결
47	14	61	울산지방법원 2015. 6. 12. 선고 2015고합52 판결
48	26	54	울산지방법원 2016. 5. 27. 선고 2015고합381 판결
49	10	112	울산지방법원 2017. 3. 24. 선고 2016고합320 판결
50	20	73	울산지방법원 2019. 10. 11. 선고 2019고합157 판결
51	40	70	울산지방법원 2019. 11. 20. 선고 2019고합132 판결
52	40	70	울산지방법원 2019. 4. 26. 선고 2018고합276 판결
53	7	441	울산지방법원 2020. 5. 29. 선고 2019고합365 판결
54	12	75	울산지방법원 2020. 8. 18. 선고 2020고합12 판결
55	12	52	의정부지방법원 2011. 10. 28. 선고 2011고합210 판결
56	10	95	의정부지방법원 2011. 4. 1. 선고 2010고합300 판결
57	8	38	의정부지방법원 2011. 4. 22. 선고 2010고합375 판결
58	71	68	의정부지방법원 2011. 4. 29. 선고 2010고합387 판결
59	13	53	의정부지방법원 2011. 5. 20. 선고 2010고합359 판결
61	10	49	의정부지방법원 2011. 9. 5. 선고 2011고합212 판결
62	12	63	의정부지방법원 2013. 5. 13. 선고 2013고합32 판결
63	27	59	의정부지방법원 2013. 5. 22. 선고 2013고합44 판결
64	39	116	의정부지방법원 2014. 3. 3. 선고 2013고합392 판결
65	11	56	의정부지방법원 2014. 8. 7. 선고 2014고합103 판결
66	10	63	의정부지방법원 2017. 2. 14. 선고 2016고합470 판결
68	11	86	인천지방법원 2015. 4. 14. 선고 2014고합856 판결
70	29	134	인천지방법원 2019. 12. 19. 선고 2019고합473 판결
72	28	72	제주지방법원 2011. 4. 18. 선고 2011고합4 판결
73	35	74	제주지방법원 2011. 5. 16. 선고 2011고합7 판결
74	5	73	제주지방법원 2015. 3. 26. 선고 2014고합243 판결
75	9	75	창원지방법원 2014. 7. 14. 선고 2014고합106 판결
76	4	77	창원지방법원 2018. 6. 25. 선고 2018고합63 판결
78	5	48	창원지방법원 진주지원 2017. 7. 18. 선고 2017고합36 판결
80	8	46	청주지방법원 2011. 8. 22. 선고 2011고합117 판결
81	16	68	청주지방법원 2013. 2. 1. 선고 2012고합330 판결
82	5	58	청주지방법원 2013. 9. 3. 선고 2013고합95 판결
83	48	52	춘천지방법원 2013. 5. 21. 선고 2013고합5 판결
84	12	68	춘천지방법원 2013. 7. 19. 선고 2013고합40 판결
85	41	65	춘천지방법원 2014. 8. 22. 선고 2014고합32 판결
86	28	91	춘천지방법원 2016. 12. 20. 선고 2016고합52 판결

<Appendix 3> List of All Sentences in Test 1-5

Setting: All [Test5]	
Tag Nr	Sentence
[156]	칼에 찔린 상처의 부위 및 개수, 깊이 등에 비추어 볼 때 이는 피고인의 주장과 같이 넘어지면서 우연히 칼날이 피해자의 가슴에 꽂혀 발생한 자상이라고 보기 어렵고,
[393]	살피건대, 형사재판에서 유죄의 인정을 저지하는 합리적 의심이라 함은 모든 의문, 불신을 포함하는 것이 아니라 논리와 경험칙에 의하여 요 증사실과 양립할 수 없는 사실의 개연성에 대한 합리적 의문을 의미하는 것으로서, 피고인에게 유리한 정황을 사실 인정과 관련하여 파악한 이성적 추론에 그 근거를 두어야 하는 것이므로,
[73]	피고인이 칼로 피해자를 찌를 당시 자신의 행위로 인하여 피해자의 사망이라는 결과가 발생할 가능성 또는 위험이 있다는 것을 충분히 예견할 수 있었음에도 이를 용인하였다고 봄이 상당하므로,
[166]	⑤ 피해자의 왼쪽 가슴 중 위에서 본 자창 주변에 칼끝에 의한 것으로 추정되는 예기 손상이 10개 가량 발견되는데,
[442]	이 사건 범행이 비록 순간적인 충동에 의하여 우발적으로 일어난 것이라고 하더라도 피고인은 자신의 행위로 인하여 피해자가 사망할 가능성 또는 위험이 있음을 인식하거나 예견하였다고 할 것이다.

Setting: No Sim_e [Test4]	
Tag Nr	Sentence
[156]	칼에 찔린 상처의 부위 및 개수, 깊이 등에 비추어 볼 때 이는 피고인의 주장과 같이 넘어지면서 우연히 칼날이 피해자의 가슴에 꽂혀 발생한 자상이라고 보기 어렵고,
[166]	⑤ 피해자의 왼쪽 가슴 중 위에서 본 자창 주변에 칼끝에 의한 것으로 추정되는 예기 손상이 10개 가량 발견되는데,
[73]	피고인이 칼로 피해자를 찌를 당시 자신의 행위로 인하여 피해자의 사망이라는 결과가 발생할 가능성 또는 위험이 있다는 것을 충분히 예견할 수 있었음에도 이를 용인하였다고 봄이 상당하므로,
[57]	피고인이 누워 있는 피해자를 뒤에서 칼로 찌른 이 사건 범행은 피해자의 피고인에 대한 현재의 부당한 침해를 방위하거나 그러한 침해를 예방하기 위한 행위로 상당한 이유가 있는 경우에 해당한다고 볼 수 없다.
[227]	자기의 행위로 타인의 사망이라는 결과를 발생시킬 만한 가능성 또는 위험이 있음을 인식하거나 예견하면 충분한 것이고 그 인식이나 예견은 확정적인 것은 물론 불확정적인 것이라도 이른바 미필적 고의로 인정된다.

Setting: No Sim_n [Test3]	
Tag Nr	Sentence
[458]	피고인이 조**, 배** 과 공모하여 이후 피해자 보험회사에 허위로 보상 접수를 한 사실도 인정할 수 있으므로,
[393]	살피건대, 형사재판에서 유죄의 인정을 저지하는 합리적 의심이라 함은 모든 의문, 불신을 포함하는 것이 아니라 논리와 경험칙에 의하여 요 증사실과 양립할 수 없는 사실의 개연성에 대한 합리적 의문을 의미하는 것으로서, 피고인에게 유리한 정황을 사실 인정과 관련하여 파악한 이성적 추론에 그 근거를 두어야 하는 것이므로,
[166]	⑤ 피해자의 왼쪽 가슴 중 위에서 본 자창 주변에 칼끝에 의한 것으로 추정되는 예기 손상이 10개 가량 발견되는데,
[73]	피고인이 칼로 피해자를 찌를 당시 자신의 행위로 인하여 피해자의 사망이라는 결과가 발생할 가능성 또는 위험이 있다는 것을 충분히 예견할 수 있었음에도 이를 용인하였다고 봄이 상당하므로,
[156]	칼에 찔린 상처의 부위 및 개수, 깊이 등에 비추어 볼 때 이는 피고인의 주장과 같이 넘어지면서 우연히 칼날이 피해자의 가슴에 꽂혀 발생한 자상이라고 보기 어렵고,

Setting: No Sim_n No Sim_e [Test2]	
Tag Nr	Sentence
[166]	⑤ 피해자의 왼쪽 가슴 중 위에서 본 자창 주변에 칼끝에 의한 것으로 추정되는 예기 손상이 10개 가량 발견되는데,
[73]	피고인이 칼로 피해자를 찌를 당시 자신의 행위로 인하여 피해자의 사망이라는 결과가 발생할 가능성 또는 위험이 있다는 것을 충분히 예견할 수 있었음에도 이를 용인하였다고 봄이 상당하므로,
[156]	칼에 찔린 상처의 부위 및 개수, 깊이 등에 비추어 볼 때 이는 피고인의 주장과 같이 넘어지면서 우연히 칼날이 피해자의 가슴에 꽂혀 발생한 자상이라고 보기 어렵고,
[227]	자기의 행위로 타인의 사망이라는 결과를 발생시킬 만한 가능성 또는 위험이 있음을 인식하거나 예견하면 충분한 것이고 그 인식이나 예견은 확정적인 것은 물론 불확정적인 것이라도 이른바 미필적 고의로 인정된다.
[228]	피고인이 범행 당시 살인의 범의는 없었고 단지 상해 또는 폭행의 범의만 있었을 뿐이라고 다투는 경우에 피고인에게 범행 당시 살인의 범의가 있었는지는 피고인이 범행에 이르게 된 경위, 범행의 동기, 준비된 흉기의 유무· 종류· 용법, 공격의 부위와 반복성, 사망의 결과 발생 가능성 정도 등 범행 전후의 객관적인 사정을 종합하여 판단하여야 한다

Setting: No Sim_n No Sim_e No Sim_g [Test1]	
Tag Nr	Sentence
[458]	피고인이 조**, 배** 과 공모하여 이후 피해자 보험회사에 허위로 보상 접수를 한 사실도 인정할 수 있으므로,
[315]	위와 같은 정신 분열증으로 인한 형법 제 10조에 규정된 심신장애의 유무 및 정도의 판단은 정신 분열증의 종류 및 정도, 범행의 동기 및 원인, 범행의 경위 및 수단과 태양, 범행 전후의 피고인의 행동, 증거 인멸 공작의 유무, 범행 및 그 전후의 상황에 관한 기억의 유무 및 정도, 반성의 빛 유무, 수사 및 공판정에서의 방어 및 변소의 방법과 태도, 정신병 발병 전의 피고인의 성격과 그 범죄와의 관련성 유무 및 정도 등을 종합하여 판단하여야 한다(대법원 1994. 5. 13. 선고 94도 581 판결 등 참조).
[313]	정신적 장애가 있는 자라고 하여도 범행 당시 정상적인 사물 판별능력이나 행위통제능력이 있었다면 심신장애로 볼 수 없음은 물론이나, 정신적 장애가 정신 분열증과 같은 고정적 정신질환의 경우에는 범행의 충동을 느끼고 범행에 이르게 된 과정에서의 범인의 의식상태가 정상인과 같아 보이는 경우에도 범행의 충동을 억제하지 못한 것이 정신질환과 연관이 있는 경우가 흔히 있고,
[329]	피고인이 위와 같은 만성화되고 조직화된 망상에서 쉽게 벗어나기는 힘들 것으로 보인다.
[55]	그렇다면 피해자의 경찰 진술에 의하여 피해자가 폭행이 끝난 후 누워 있는 상태에서 피고인이 피해자를 칼로 찌른 이 부분 공소사실을 충분히 인정할 수 있다.

<Appendix 4> Limitation of Alternative Hypothesis Retrieval

Case 1: Non-specific query data

Query-left	범행 당시 피고인이 술에 취했다는 점은 인정되나 (rsupport)
Query-right	피고인이 이 사건 범행 당시 심신미약 상태에 있었다고 주장한다. (rebuttal)

Setting: All Similarity	
Tag Nr	Sentence (warrant - backing)
[194]	한편 피해자는 당시 어떠한 반항도 하고 있는 않은 상황이었으므로 피고인의 실수로 칼이 빗나가 위 부위에 찔렸다고 보기도 어려운 점 등에 비추어 보면,
[54]	또한, 피해자가 목 뒷부분에 상처를 입었다는 점에서 서로 마주 보고 앉은 상태에서 실랑이를 벌이다가 피해자의 목 뒷부분을 베었다는 피고인의 주장은 믿기 어렵다.
[76]	(1) 형법 제 21조 제 2 항, 제 3 항의 면책적 과잉 방위가 성립하기 위해서는 방위행위가 상당성을 초과한 경우로서 그 행위가 야간 기타 불안스러운 상태에서 공포, 경악, 당황, 흥분 등 행위자의 열악함에서 나오는 심약적 충동에서 비롯된 것이어야 한다.
[92]	피고인의 행위는 자의에 의한 것이 아니라,
[57]	피고인이 누워 있는 피해자를 뒤에서 칼로 찌른 이 사건 범행은 피해자의 피고인에 대한 현재의 부당한 침해를 방위하거나 그러한 침해를 예방하기 위한 행위로 상당한 이유가 있는 경우에 해당한다고 볼 수 없다.

Setting: No Similarity	
Tag Nr	Sentence (warrant - backing)
[57]	피고인이 누워 있는 피해자를 뒤에서 칼로 찌른 이 사건 범행은 피해자의 피고인에 대한 현재의 부당한 침해를 방위하거나 그러한 침해를 예방하기 위한 행위로 상당한 이유가 있는 경우에 해당한다고 볼 수 없다.
[140]	피해자와 피고인 사이에 몸싸움이 벌어졌을 때 회칼을 사용해야 할 정도로, 피고인이 피해자에게 제압되어 다른 수단을 강구할 수 없을 만큼 급박한 상황이었던 것으로 보이지는 않는 점,
[199]	자기의 행위로 인하여 타인의 사망의 결과를 발생시킬 만한 가능성 또는 위험이 있음을 인식하거나 예견하면 되고
[93]	피해자의 말에 따라 자신의 범행을 은폐하기 위한 방편으로써 그렇게 한 데 불과하고,
[74]	이 사건 범행 당시 피고인에게는 적어도 살인의 미필적 고의가 있었음이 충분히 인정된다.

Case 2: Warrant-claim as query

Query-left	피고인이 피해자의 죽음을 충분히 예견할 수 있었고, 따라서 살인의 미필적 고의가 있었다고 할 수 있으므로 (warrant)
Query-right	피고인의 주장을 받아들이지 않는다. (claim)

Setting: All Similarity	
Tag Nr	Sentence (rebuttal-rsupport)
[351]	그러나 피고인이 사무실에서 나갈 때까지만 하더라도 피해자는 살아 있었는 바, 피고인은 피해자를 살해하거나 피해자의 사망에 관여한 사실이 없다.
[391]	4) 피고인 및 변호인은 피고인이 소지하고 있던 이 사건 해머, 검정색 코팅 장갑, 흰색 긴팔 와이셔츠가 2013. 9. 11. 피고인의 주거지 부근에서 발견된 것과 관련하여,
[392]	누군가가 위 물건들을 가져간 후 피해자의 혈흔을 묻혀 피고인의 주거지 부근에 놓는 등의 방식으로 사건을 조작하였을 가능성이 있다는 의문을 제기한다.
[359]	2) 피고인 및 변호인은 피해자가 물품 창고에서 의자 위에 올라 서서 작업을 하다가 의자에서 떨어지면서 피해자 주변에 있던 둔기 유사의 물체에 뒤통수 부위를 부딪혀 머리 손상이 발생하였을 가능성이 있다고 주장 하나,
[304]	검사는, 피고인이 자신에 대한 국립 중앙도 서관 측의 대우에 화가 나 이 사건 범행을 저지른 점,

Setting: No Similarity	
Tag Nr	Sentence (rebuttal-rsupport)
[266]	○○ 호 순찰차에 타고 있던 경찰관 내지 다른 경찰관이 쏜 총알에 피해자 C가 맞았을 가능성을 배제할 수 없다.
[373]	피해자의 두개골이 함몰된 면적도 이 사건 해머 머리 부분의 크기(지름 약 6cm)에 비해서 작다는 이유로 피해자가 이 사건 해머에 머리를 맞아 사망하였을 가능성은 희박하다고 주장한다.
[255]	즉, 피고인에게 애당초 피해자 C를 죽일 의도가 없었다.
[273]	피고인은 ' 사람을 죽이는데 도와 달라' 는 이○○ 의 말을, 피해자들을 상해 하는 데에 도움을 달라는 부탁으로 받아들여 피해자들을 태운 승용차를 이 사건 범행장소까지 운전해 간 다음 상해를 가할 의사로 피해자 박○○ 의 다리를 야구 방망이로 몇 대 때린 적이 있을 뿐이지,
[48]	서로 마주 보고 앉은 상태에서 피고인이 칼을 든 피해자의 왼손을 피고인의 양손으로 잡고 실랑이를 벌이다가 피해자의 목 뒷부분을 벤 것으로,