국제학 석사 학위논문

Transformer-based Legal Argument Structure Extraction Model for Crime Investigation Analysis

범죄수사 분석을 위한 트랜스포머(Transformer) 모델 기반 법적 논증 구조 추출 모델에 관한 연구

구 예 리(Gu, Ye Ri)

국제학과(Department of International Studies)

정보법과학전공(Major in Legal Informatics and Forensic Science)

한림대학교 대학원

(Graduate School, Hallym University)

국제학석사학위논문

Transformer-based Legal Argument Structure Extraction Model for Crime Investigation Analysis

2022

구혜리

국제학 석사 학위논문

Transformer-based Legal Argument Structure Extraction Model for Crime Investigation Analysis

범죄수사 분석을 위한 트랜스포머(Transformer) 모델 기반 법적 논증 구조 추출 모델에 관한 연구

구 예 리(Gu, Ye Ri)

국제학과(Department of International Studies)

정보법과학전공(Major in Legal Informatics and Forensic Science)

한림대학교 대학원

(Graduate School, Hallym University)

박 노 섭, 노 기 영 교수지도

국 제 학 석사 학위논문

구 예 리의 석사학위논문을 합격으로 판정함

2022 년 12 월 20 일

심사위원장 장 윤 식

심사위원 박 노 섭

심사위원 노 기 영

심사위원 안 정 민

Table of Contents

List of Tables

List of Figures

I. Introduction

1. Overview

Legal Artificial Intelligence (Legal AI) is a research that focuses on the application of artificial intelligence to help process legal tasks. Most of the data used in legal domains are expressed in texts. Therefore, the tasks of legal AI are mainly dependent on Natural Language Processing (NLP) technology.

The application of Legal AI can play an important role in reducing the repetitive work of legal professionals [1]. The majority of tasks that exist in the legal domain require the expertise of legal professionals and a complete understanding of various legal documents. Therefore, a well-devised Legal AI system can reduce the time spent on redundant tasks, thereby contributing to the development of the legal field. Moreover, Legal AI can be used to provide reliable legal knowledge not only to the experts but also to the general public who may not be familiar with legal issues. Due to such advantages, the movement to apply artificial intelligence to the field of law is steadily increasing, as well as within police investigations.

In particular, with the recent amendments to the Korean Criminal Procedure Act in 2021 that focus on the reviewing process in police investigation, the importance of developing an AI-assisted investigation support system has been emphasized. The revised act supports court-oriented trials by granting the authority to close cases without sending them to the prosecutor(Article 312 of the Amended South Korean Criminal Procedure Act)[1] as well as limiting the admissibility of the suspect interrogation reports from the prosecution[2] (Article

[1] The police are given the right to conduct primary investigations, abolish the prosecutor's pre-delivery investigation command, and prepare an objection procedure for rejecting a warrant, and strengthen the responsibility and completeness of the police investigation by granting the right to terminate the primary investigation ("Adjustment of the investigation process of the prosecution and the police", Korea Policy Briefing, 2020.08.25)

[2] Article 312 (Protocol Prepared by Prosecutor or Senior Judicial Police Officer) A protocol

245). This movement toward Court-Oriented Trials further highlights the need for evidence-based logical argumentation for police by deviating from the documents prepared by investigative agencies and demands the verification of the investigation grounded on logicality.

However, most of the existing case analysis assistance tools focus on the acquisition and analysis of the evidence rather than logical verification, thus preparing a legal argument analysis system able to derive logical claims from evidence is required to cope with the changes in the investigation environment under the revised Criminal Procedure Act [2].

Historically, argumentation has been considered an essential area in philosophy, and with recent advances in technology, their relevance has grown exponentially in other domains including logic, law, and artificial intelligence [3]. The purpose of argumentation is to persuade others to accept a view of a particular claim and aims to draw conclusions from a premise that is acceptable to everyone. Therefore, verifying criminal investigation based on argumentation can be considered an essential step to ensure objective and uniform quality of investigation with logical completeness.

However, the amount of evidence to be collected and the difficulty of analysis is rapidly increasing due to the accelerated completeness of crime in Korea, while the lack of human and material resources[3] makes it inevitable to vary the quality of investigation. Therefore, it is required to improve the completeness of the investigation by deriving objective and homogeneous results through the development of an argument-based verification system for the investigation. Furthermore, it is necessary to study how to logically verify the investigation results and quickly analyze complex cases through visualization as a means of responding to logical attacks that may be presented by lawyers in court.

---

concerning interrogation of a criminal suspect, prepared by a prosecutor shall be admissible as evidence, only when it was prepared in compliance with the due process and proper methods and the criminal defendant, who was the suspect at the time, or his or her defense counsel admits its contents at a preparatory hearing or a trial. <Amended on Feb. 4, 2020>

[3] The average number of cases a police investigator takes charge of over a year is 84.5 cases in 2020, and the average population per police officer by local government is 411 nationwide (National Police Agency, 2020 Police Statistical Yearbook)

2. Problem Statement

With the rapid advancement of technology, the number of documents that need to be processed by police investigators has been growing exponentially. Manually analyzing this vast amount of information presents a challenge as the process may be tedious and time-consuming thus slowing the investigation process. To overcome this issue, various tools aiding the investigation process have been developed including Sandbox [4] and Aruvi [5]. The tools aim to help users to structure their logic by supporting the construction and visualization of the arguments with graphs or diagrams. Using such software allow users to strengthen their arguments by revealing logical gaps and inconsistencies. While some of these tools allow users to employ underlying logical theories to build their arguments [6], most of the current argument structuring tools still possess limitations as they do not provide automated analysis [7], [8].

Argument structuring is effective in that it allows investigators to explicitly express each step of the argumentation and identify the strengths and weaknesses thereby understanding the logical connectivity [9]. Despite its usefulness, relying on pen and paper to structure the reasoning has been considered laborious [10]. Especially for the legal domain, it is necessary to build an automated system to identify and structure the arguments as it is one of the most refined fields of argumentation. Therefore, in this study, we focus on automating the extraction of argument structure from case-related documents using Natural Language Processing (NLP) techniques, known as argument mining, to help maintain the completeness of legal logic.

This research also investigates the application of Transformers [11] in argument mining tasks. The development of Transformers has led to remarkable performance gains in various domains by fine-tuning the large pre-trained language models such as BERT [12] on different tasks. However, little has been studied on the application of Transformer-based architectures in the field of argument mining, especially in the Korean language. Hence, our study aims to use Transformers on our dataset to improve performance.

3. Objectives

This research aims to provide an automated argument structure extraction system that can accelerate and improve the crime investigation process. For this, we approach this by focusing on the three subtasks following the previous works that tackled a similar problem. Argument component identification aims to distinguish the argumentative role of the text according to the argumentation model [13], [14], [15]. Argument relation detection involves the identification of the relationships between the argument components, whether one supports or attacks the other or is not related to it [14], [15], [16]. For argument structure extraction, the goal is to identify the argumentation pattern and visualize them in graphs [17], [18].

The table below shows the sub-tasks and the corresponding objectives we defined in this research.

Table 1 Tasks and Following Objectives of the Research

Task	Objective
Corpus Creation	Build a reliable argumentation corpus annotated using an argument model devised for crime analysis
Automatic Argument Component Identification	Apply transformers to the argument component identification task by fine-tuning BERT-based models
Automatic Argument Relation Detection	Apply transformers to detect the argument relationships by fine-tuning BERT-based models
Automatic Argument Structure Extraction	Extract argument structures from the automatically annotated data and visualize them as graphs

4. Thesis Outline

This thesis is divided into 5 chapters. Chapter II presents an overview of the theoretical concepts of argumentation and the algorithms and techniques used to represent texts to handle them computationally. We also explain the usage of argument mining in various fields of study including the legal domain.

Chapter III provides a description of the corpus used in this study and the process of annotation and the final analysis of the created dataset.

Chapter IV presents the architecture of the proposed model for the tasks of this study. For argument component identification, we use a pre-trained bi-directional transformer to automatically classify the argument components from texts. For argument relation identification, we use a multiple-choice classifier to choose a related argument pair from the text and classify their relationships using a Natural Language Inference model which can be used to extract the argumentative structure of the document.

Chapter V shows the results of the experiments and its analysis. For argument component identification, we provide the performance scores of our proposed model and compare them with the baseline model. For argument relation identification, we evaluate the model using various metrics and provide a detailed analysis of the misclassified data. Lastly, we show the result of the argument structure extraction model by testing it with a sample court decision.

Finally, in Chapter VI, we discuss the achievements and limitations of the study and present ideas for extensions of this thesis in the future.

II. Literature Review

1. Argumentations Theory

A. Argumentation Models

The study of argumentation is a highly interdisciplinary field that involves discussion from various domains including philosophy, law, communication, psychology, and artificial intelligence [13]. The investigation of argumentation started with Aristotle's works in the 6th century B.C by defining the theory of logical reasoning and argumentation [19]. Branching off from Aristotle's theory, various studies have defined argumentation. Ketcham [20] defines argumentation *as the art of persuading others to think or act in a definite way. It includes all writing and speaking which is persuasive in form.* Fox et. al describes arguments as tentative evidence of the proposition [21]. Overall, while there is no unitary definition, the consensus appears to be that the purpose of argumentation is to persuade others [22].

During the process of argumentation, arguments are interchanged to support an idea, referred to as a *claim*, through logical reasoning and offering evidence(e.g. facts) to convince that the *claim* is the legitimately derived *conclusion* from the given arguments. The claim can be used as a premise for another claim as well, and create a chain of reasoning [13]. The study of argumentation is crucial in many areas which require human reasoning mechanisms including legal domains and artificial intelligence, as the ability to formulate persuasive arguments plays an important role in analyzing the overall decision-making process and the different stances [23]. For a better understanding of argumentation, the components and their relations are often analyzed which are represented either via natural language or diagrams. Argument diagrams establish the components of arguments and visualize their

respective relationships. Thus far, several approaches and models have been developed for structuring argumentation [8].

1) Argumentation Structures

In order to represent the argument structure, the argument diagram is often used. The argument diagram consists of two basic elements [24]. A set of circled numbers representing a proposition (premise or conclusion) is connected by lines or arrows where each line (arrow) represents an inference. This network of points and lines presents an overview of the reasoning in a given argument, showing various premises and conclusions [25].

① Whatley

The first example of the argument diagram used to describe the argument process can go back to Richard Whatley in 1836. In his textbook 'Elements of Logic', he explained that he takes a 'train of claims to us' and reduces arguments to a form that can be applied to logical rules [26]. This approach finds the conclusions of the argument and traces the reasoning to find out the basis for the argument [24, p.421]. This process is repeated and can search for an additional basis for the premise [24, p.422]. The figure below shows the diagram of the ‘chain of arguments' he described.

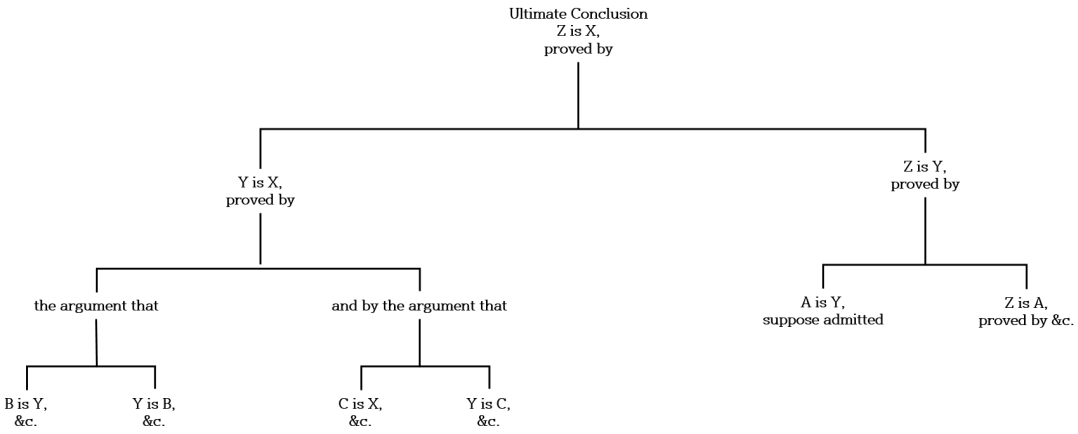


Figure 1 Whately’s diagramming (Whately, 1836, p. 422)

The modern approach of most argument diagramming methods is rooted in Whately's style, which graphs the link between the premise and the conclusion [27]. In his diagram, each argument is represented as a node connected by lines to construct graphs in forms such as a tree, where the root node is labeled as the 'Ultimate Conclusion'. Each link in these argument chains is a conclusion supported by a premise in the following steps of arguments [25].

② Beardsley

After Whatley, Beardsley's diagrammatic summary was introduced describing the basic types of argument structures and how they were constructed. In his book *Practical Logic* [28], he used circled numbers to represent statements as nodes and arrows to join the nodes. This structure is defined as the 'skeletal pattern' of the argument [28], He formulated several important principles of argument diagramming, namely the Rule of Grouping (keeping the reasons for a conclusion close to each other), or the Rule of Direction (maintaining the direction of a serial argument in one-way) [25].

2) Legal Argumentation Structures

③ Wigmore

In 1917, Wigmore was the first to introduce the method to visually represent legal evidence in diagrams, known as the charting method [29]. He attempted to understand a large body of evidence by using diagramming to map evidence and inferential links put forward by both parties in a trial.

The goal of Wigmore's charting method is to prove the acceptability of a hypothesis for a given evidence. In his evidence chart, the argumentation is drawn as a tree graph. The root node of this chart is the central charge in a case proven by prosecutors or refuted by lawyers, while arrows represent inferences. [8]. In his chart, the following four types of evidence are distinguished and represented as distinctive symbols:

1) Testimonial evidence refers to testimony introduced by a witness and represented as squares (nodes 1 and 7, 9, 11, 13 in Figure 2).

2) Circumstantial evidence is deduced from other facts and uses circles to display them (nodes 2 and 4, 6, 8 in Figure 2).

3) Corroborative evidence is used to support or reinforce the root nodes or inferences. They are introduced as triangles (nodes 10 and 12 in Figure 2).

4) Explanatory evidence demonstrates circumstantial evidence and refutes testimonial evidence. An angle is used to represent them (node 3 in Figure 2).

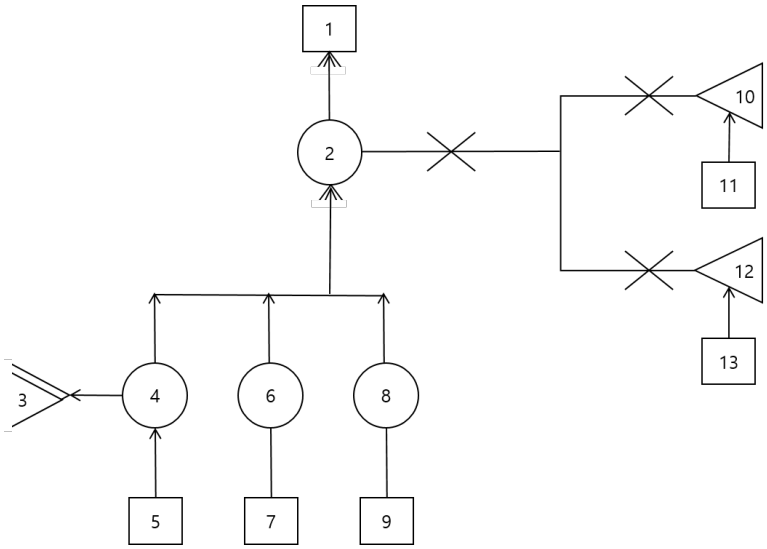


Figure 2 A Sample Wigmore Diagram

The lines between the nodes are used to represent the reliability of the evidence according to their respective shapes, with a double arrow or X indicating strong support, and arrowless lines indicating an average degree of support.

④ Toulmin

Toulmin's argument model was introduced in 1958 in his work, The Uses of Argument. He developed a simple six-part structure diagram for understanding reasoning found in jurisprudence, which has since become a popular argumentation model [30]. His argument model consists of six components, namely, datum, warrant, backing, qualifier, rebuttal, claim. With claims being the conclusion of the argumentation, datums are facts that lead to claims through inferences. Warrants refer to the logical bridge that connects the gap between

datum and claim thus justifying the inference from datum to claim. The acceptability of warrants is shown by backings that correspond to statements of facts. However, rebuttals dismiss the authority of the conclusion by attacking the link between datum and claim. Finally, qualifiers are also placed between the datum and claim showing the strength of support with warrants. The argumentation diagram layout proposed by Toulmin has been used in other studies [22], [24], [27]. Figure 3 demonstrates an example of the Toulmin argumentation model.

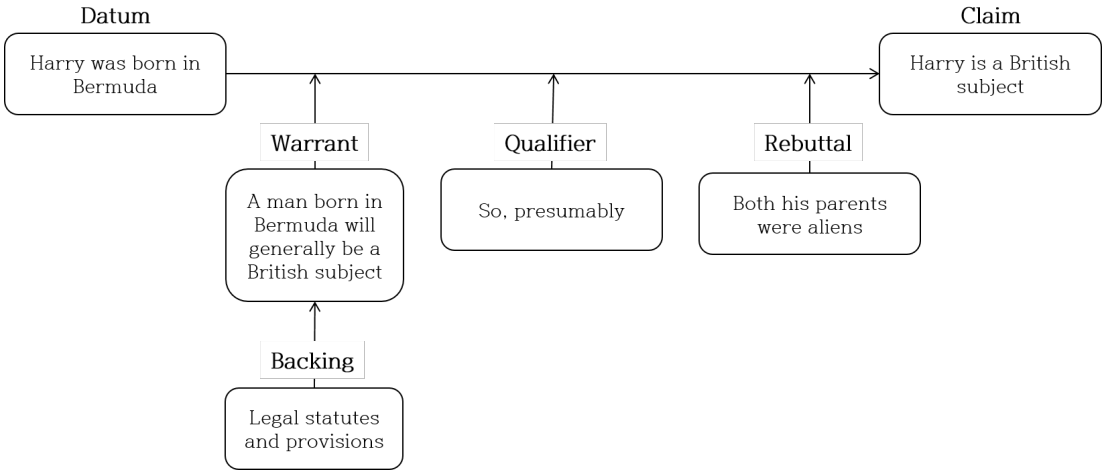


Figure 3 Toulmin's Argument Model

2. Text Representation

In order to use argumentation as data, it is necessary to comprehend the methods to treat text computationally. As a source of information, text data holds valuable insights that cannot be obtained from analyzing quantitative data [31]. The computational techniques for text analysis and representation are called Natural Language Processing (NLP). NLP aims to achieve human-like language understanding for varying applications. Previous studies have combined machine learning with NLP to execute specific tasks such as machine translation, information retrieval, text analytics, decision-making, and information visualization [32].

Text can be viewed as a set of entities at various granularities, such as documents, sentences, words, or characters. Most NLP algorithms employ a variety of methods to infer vectors using implicit relationships between texts. The objectives of these methods are to represent unstructured text in a form suitable for machine learning to treat computationally which then can be applied to tasks such as clustering, dimensionality reduction, or text classification [33].

Generally, for text classification, two main steps are taken to process the unstructured text: Text preprocessing and Feature extraction. After this, the learned representation can be used for classification using an appropriate classifier [34] [35].

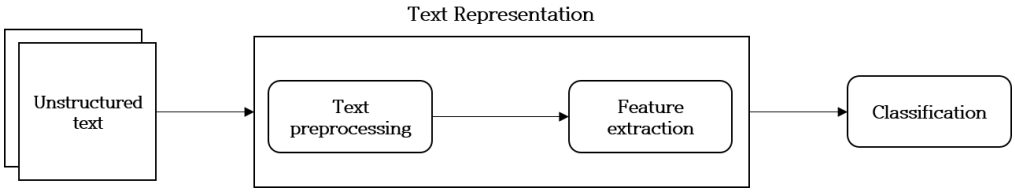


Figure 4 Text Classification Pipeline

A. Text Pre-processing

Given an input text, the first module in an NLP pipeline is a tokenizer that transforms texts into sequences of words [36]. A tokenizer splits texts into words, phrases, or other meaningful units as necessary. The individual tokens serve as input for various machine-learning models. Depending on the desired preprocessed result, preprocessing can be divided into low preprocessing and high preprocessing. While low-level preprocessing is related to tasks such as sentence boundary detection, part-of-speech tagging, and noun phrase chunking, high-level processing deals with processing at the semantic level including name entity recognition, relation extraction, and temporal extraction [37].

To filter words without critical significance and are present in high frequency from the text (e.g., conjunctions and prepositions), such stopwords are removed to retrieve more accurate features.

B. Feature Extraction

After preprocessing, features are extracted from the text. Features refer to individual characteristics that represent the data such as symbols or numerical values [37]. Extracting features from data is the process of converting raw data into a numerical format enabling a human-like understanding of classifiers [33]. Therefore, choosing informative and discriminative features is a critical element for the effective training of the classifier. Here, we introduce various feature extraction methods adopted in previous works on NLP tasks.

1) Word Representation

Word representation is a process that transforms symbols into machine-understandable meanings. It aims to numerically represent words by reflecting linguistic characteristics so that text can be applied to models for natural language processing. When quantifying words, they are transformed into a vector containing the frequency of the words in a text [33]. Therefore, word representation is expressed as word embedding or word vector.

① One hot encoding

The most simple and direct way to represent text is one hot encoding. Using this method, the number of dimensions is the same as the number of terms that exist in the vocabulary. Since every term in the vocabulary is represented by binary values such as 0 or 1, every word is assigned to a dictionary of the same length [33].

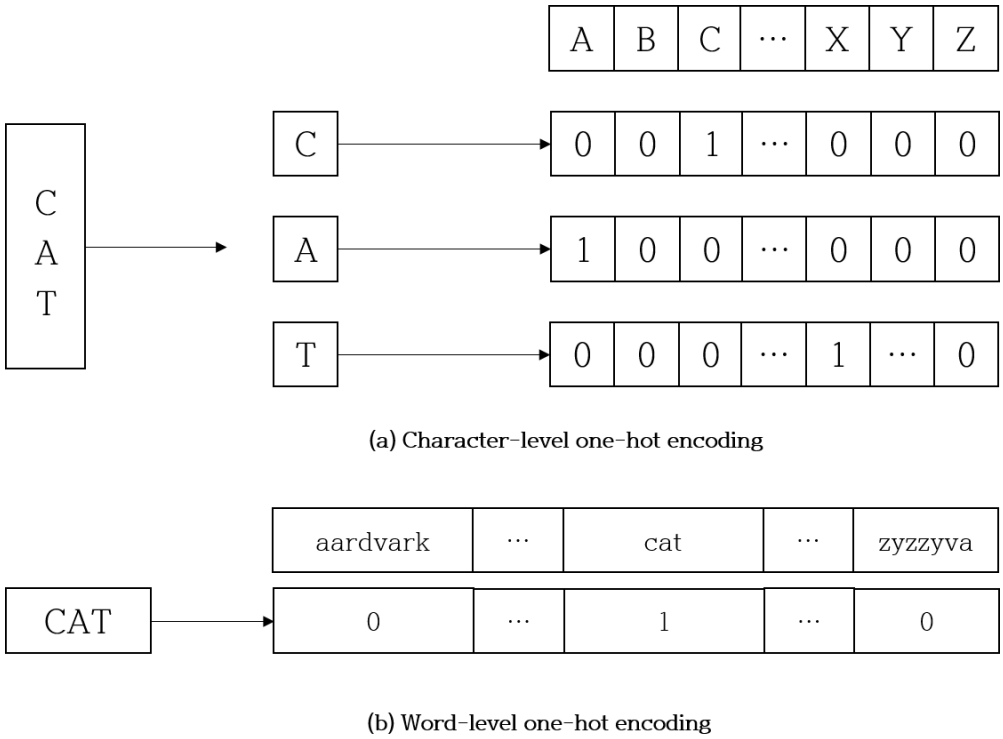


Figure 5 One-hot representation in two different ways

② Bag-of-Words (BoW)

BOW is an extension of one-hot encoding that aims to extract features from the unstructured text for machine learning algorithms [38]. A matrix of words generated using BOW ignores the semantic relationship between words as well as the grammar and order of words. As BOW encodes every token of the vocabulary as a one-hot vector, the increased size of the vocabulary can induce a sparse matrix containing a large number of "0s" without information about the order of text and grammar in the sentence [33].

③ Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) is the most common method for measuring the terms' weights in a vector space model. TF-IDF was presented by [39] for text representation to reduce the impact of commonly appearing words in the corpus.

TF represents the term frequency and IDF is the inverse document frequency

used to reduce the influence of frequently appearing words. Unlike TF, IDF gives more weight to words with higher or lower frequency [33]. TF-IDF is mathematically expressed by the following equation.

$$TF - IDF(t, d, D) = TF(t, d) \times log\left(\frac{D}{df_t}\right)$$

Here, t and d each represent the term and document, D is the collection of documents and $d\ ft$ is the sum of documents with the term t in it.

As TF-IDF is based on the concept of BOW, the order of words or the context is not captured by the model. The model also perceives similar expressions such as synonyms as completely different words. Thus, it is recommended to use TF-IDF as a lexical-level feature [33].

2) Word Embedding

Representing words based on their frequency raised the need for continuous vector space representation of words as they cannot capture the syntactic and semantic meaning of the words that can be utilized by models [40]. Especially, since the advent of neural network models that are capable of discovering word representation, the traditional feature extraction methods have been changing. Word representation can be learned by using supervised or unsupervised methods and for NLP tasks, unsupervised word representation methods such as word embeddings have been replacing the traditional representation approaches [33].

Word embeddings are word representation vectors that map words from the vocabulary as vectors thus creating correlations between relative and semantic similarities [33] [40]. These word embeddings capture the meaning of words without losing the order of words and are pre-trained by predicting the words thus helping various NLP tasks. Word embeddings are effective compared to previous word representation methods in that they maintain the semantic similarity of context and use low-dimensional vectors. These attributes of word embeddings contribute to its wide use in many different applications [33].

Some of the popular word embedding methods such as Word2Vec and ELMo are discussed here.

① Word2Vec

Word2vec is a word embedding that can represent the relation between similar words developed by [41]. The main idea of the model is words can have “multiple degrees of similarity”, which enables similar words to be found in a subspace of the original vector space [42]. Word2vec generates embedding vectors using Common Bag Of Words (CBOW) and Skip Gram. The difference between the two models lies in the input and the predicted results. While CBOW takes the context of a word aiming to predict the correct word based on the given context, Skip Gram takes a single word as input and predicts the relevant context.

Context windows are used to predict the target, and the range of windows is determined by changing the choice of surrounding words and the target which is a method referred to as sliding windows. The figure below shows the reversed architecture of the two models.

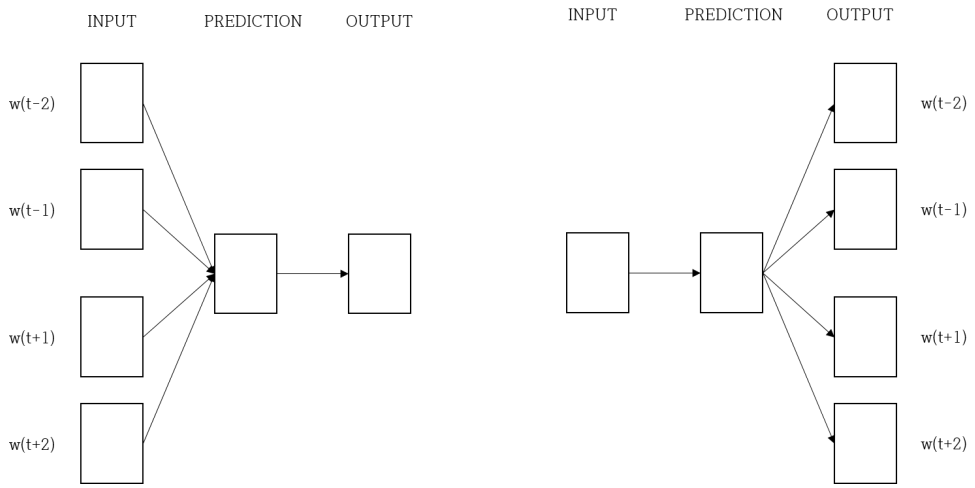


Figure 6 CBOW and Skip-gram Example

② Contextualized Word Embedding

However, modeling complex characteristics of word use and their linguistic context can be challenging when using word2vec since each word is only represented in a single vector which fails to capture its contextual meaning when the word has more than one meaning (i.e., polysemy). To compensate for the shortcomings of Word2Vec, contextualized word embedding has been devised to express not only the meaning of words but also information about the context.

Contextualized word embedding allows each word to have a different embedding depending on what it means in the context. One of the popular algorithms using contextualized word embedding is ELMo.

i. Embeddings from Language Models (ELMo)

Embedding from Language Models (ELMo) is a deep contextualized word representation method devised by Peters et al. [43] The core concept of ELMo is that it uses a pre-trained language model. Language modeling aims to predict the next word in a sentence using the previous word. Unlike the unidirectional RNN language model which reflects the contextual information of sentences sequentially, the bi-directional model used in ELMo predicts not only the following words but also the previous words by training on language models in both directions. Thus, designing a model to generate word embeddings in a such way makes it possible for words to have different embeddings depending on the context. The architecture of the model is shown below.

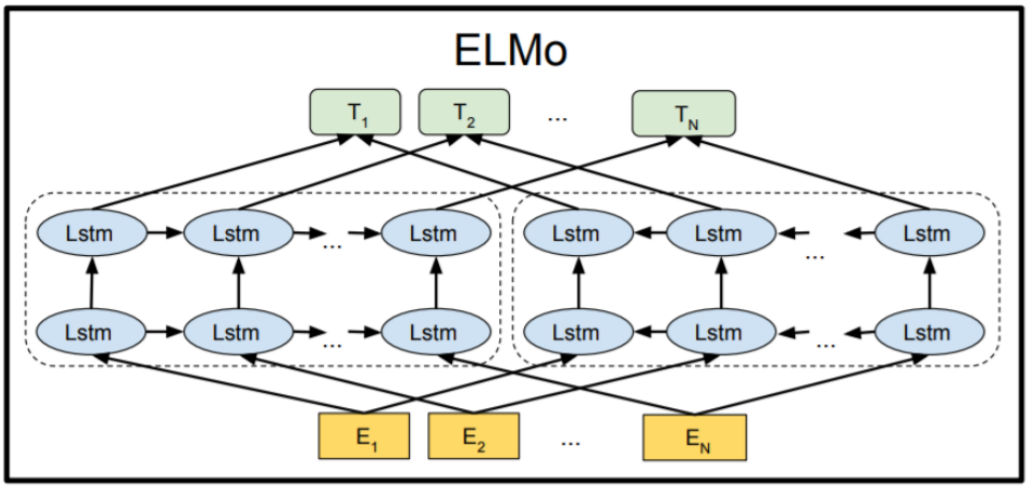


Figure 7 An Illustration of ELMo Model [43]

The authors showed that ELMo can be trained on various downstream tasks e.g. question and answering, textual entailment, and semantic role labeling.

C. Classification Methods

1) Support Vector Machines (SVMs)

The Support Vector Machines (SVMs) [44] model is a vector space-based machine learning method widely used for its strength in text classification tasks [45]. The model aims to find the optimal decision boundary between two classes that maximizes the margin for classification from any point in the given training data [46].

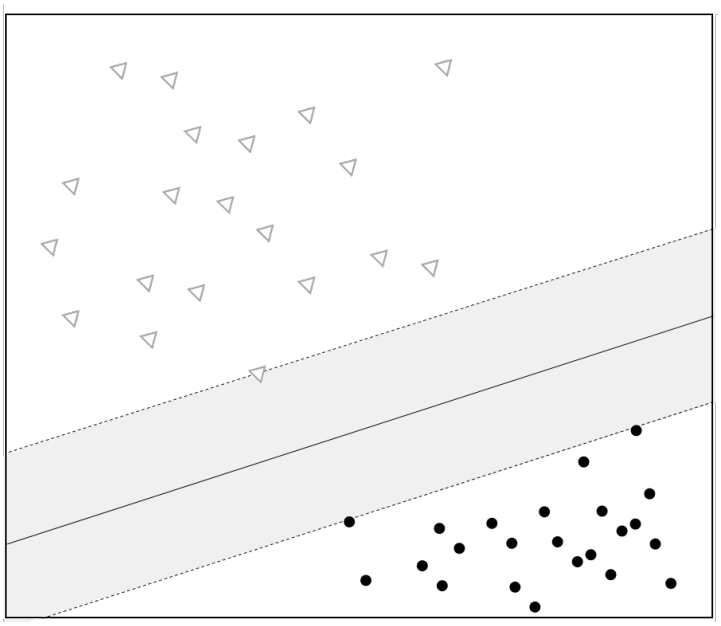


Figure 8 An Example of SVM Classification

SVMs are binary classifiers, however, to extend SVMs on multi-class classification tasks, the Pairwise approach and One-vs-Rest(OVR) approach are commonly used [47]. The pairwise approach trains a separate binary classifier for individual class pairs and their outputs are combined to predict the classes. The OVR approach trains the classifiers for the same number of given classes and chooses the final class which classifies the test data by the largest margin [46].

2) Naïve Bayes

A Naïve Bayes classifier is a probabilistic learning method popular for its simplicity and efficiency which assumes that the individual features are mutually independent of the class [48]. The probabilistic model of Naive Bayes classifiers

is based on Bayes' theorem, which can be represented as the following formula:

$$P(c_i|x) = \frac{P(c_i)P(x|c_i)}{P(x)}$$

where $P(c_i)$ is prior information on the probability of class c_i occurring, $P(x)$ is the observed information, which is the knowledge obtained from the text itself to be classified, and $P(x|c_i)$ is the likelihood of document x belonging to class space. A vector of variables $x = x_j$ refers to a document where x_j are features found in the text x and $c = \{c_1, c_2, \ldots, c_i\}$ is the set of class labels. Text classification task corresponds to assigning a class label c_i, to a document [49].

Bayes classifiers incorporate this information to calculate the Maximum a Posteriori (MAP), where document x belongs to each class $P(c_i)$ and assign the document to the class with the highest probability [49], which can be formulated as

$$\hat{c}(x) = arg\ max_i\ P(c_i|x)$$

As the Naïve Bayes classifier assumes the components of x are to be independent of each other, the likelihood can be expressed as below.

$$P(x|c_i) = \prod_j P(x_j|c_i)$$

Therefore, the predicted class $\hat{c}$ can be written as below which can be used as a measure of the amount of evidence for the documents in the class [46].

$$\hat{c}(x) = arg\ max_i\ P(c_i) \prod_j P(x_j|c_i)$$

One of the variations of this model is called the Multinomial Naive Bayes classifier [50] is a widely used algorithm for text categorization tasks. For this model, the number of occurrences of each feature is represented in the feature vector [51].

D. Deep Learning-based Methods

1) Recurrent Neural Network

Recurrent Neural Networks (RNNs) improve traditional language models with many limitations in remembering previous words by learning all previous words in the corpus. RNNs are neural networks where nodes in the hidden layer are

connected with directions to form a recurrent structure, thus using the information in the previous state to predict the information in the next state. This structure makes RNNs effective for sequential data such as texts as it maintains its order. The h_t vector of an RNN is calculated with the neighbor unit (h_{t-1}) and the input x_t using the following formula.

$$h_t = f(Wx_t + Uh_{t-1} + b)$$

However, the most common issues with RNNs are gradient vanishing and exploding problems [52]. In theory, the h_t vector stores all the information from the previous state. Nonetheless, the vanishing gradient problem, one of the major drawbacks of RNNs, happens when the sequence is long and therefore creates deeply layered neural networks and degrades the model's performance by not updating the parameters properly. The basic architecture of RNNs is shown in the figure below.

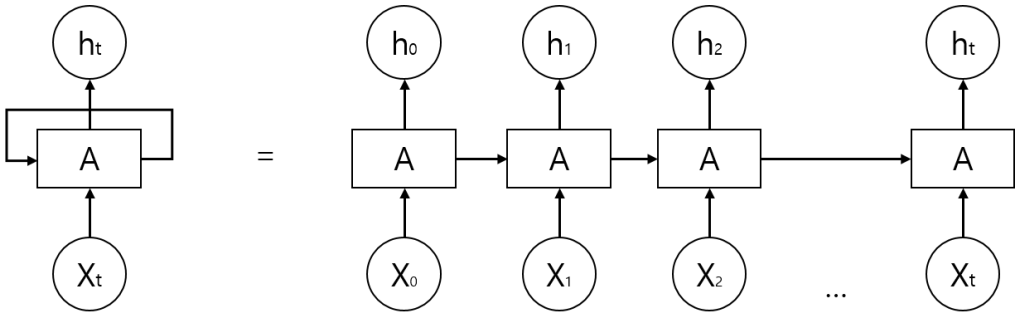


Figure 9 Basic Structure of RNNs

2) LSTM

One solution to the problem of RNNs is called long short-term memory (LSTM) networks invented in 1997. LSTM follows the basic structure of RNNs while focusing on the cell state. LSTM uses 3 gates referred to as input, forget, and output gates in the memory cell of hidden states that control the cell states consisting of a sigmoid layer and pointwise operation. The gates decide to disregard or keep the information based on the output (0 or 1) from the sigmoid layer.

E. Transformer-based Pre-trained Language Models

The transformer was first introduced in 2017 and became a popular

architecture in NLP due to its efficiency and speed [11]. A transformer model consists of an encoder that represents the original text on a deep learning space, and a decoder that generates the next word using the original text and previous output results. Encoder and decoder are largely composed of modules that use self-attention mechanisms to select context words that are important for predicting masked words or previous words that are important for predicting next words, and another module that calculates deep semantic expressions using two layers of feed-forward neural network. The architecture of a transformer model is shown in the figure below.

Unlike conventional methods such as CNN and RNN, a transformer model can directly calculate the relationship between multiple words that have important relationships with each other to reflect them and can easily parallelize. Due to these advantages, Transformer is being widely applied in the field of natural language processing. Therefore, here we discuss the techniques used in the transformer and review transformer-based models.

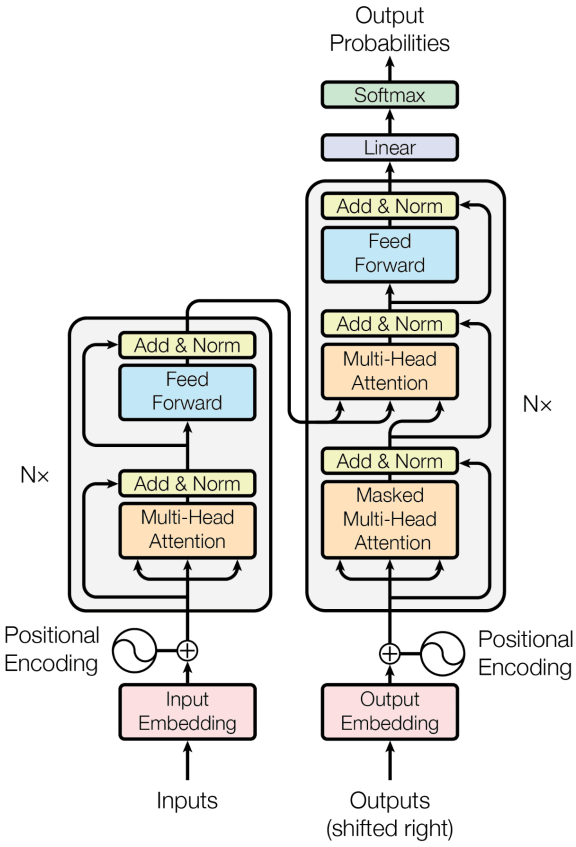


Figure 10 An Architecture of Transformer Model [11]

1) Seq-2-Seq Model

Sequence-to-sequence (seq2seq) is a model that transforms sequences from other domains from the input sequence. It is used in various fields such as machine translation translating text to another language [53] and question answering [54]. A sequence-to-sequence model consists of two separate LSTM-based models each called Encoder and Decoder, thus it is an encoder-decoder model. The encoder process the input sequence to create a context vector in the form of a hidden state vector containing the context captured by the encoder. This vector is sent to the decoder which then formulates the output sequence.

In the training phase of the decoder, the input is the context vector and the input

sequence starts with the special token [SOS]. However, in the prediction phase, the decoder input is the context vector and the [SOS] token, which then generates a sequence that starts with the [SOS] token and adds the predicted output to its input. This token generation step is repeated until the [EOS] token indicates the end of the sequence.

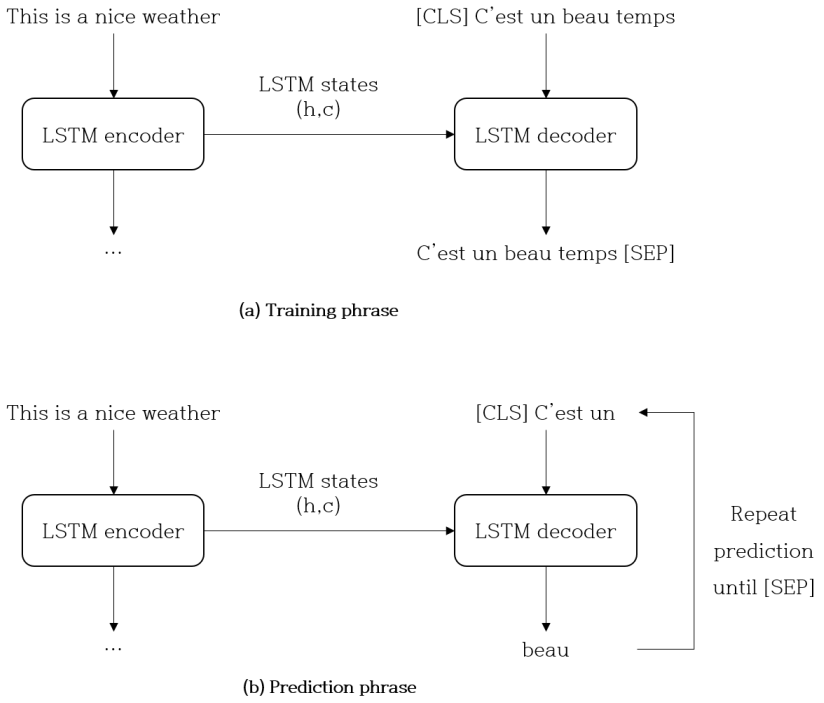


Figure 11 Seq-2-Seq Model Architechure

The seq2seq model's main limitation is the information loss from creating a fixed-sized context vector regardless of the length of the input.

2) Attention

To overcome the limitation of classic seq2seq models, a solution was proposed by [11] where the concept of "Attention" was introduced. This technique has significantly improved the machine translation system by reflecting every token of the input sequence. The core idea of attention is to refer to the encoder's entire input sentence at every time step of the decoder's prediction of the output token. However, instead of referring to the input sequence in the same proportion, it pays more 'attention' to the input token related to the token to be predicted at

that time step.

When expressing attention, it can be expressed as the following calculation.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Here, Q is a query matrix where a word in a sequence is represented as a vector and K is the matrix of all keys which refers to the vectorized representation of all tokens in a sequence. V corresponds to a value, an expression of a token and d_k is the dimension of the key. The $softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$ shows how the query affects the keys and the qV is the attention which shows how Q in the context of K modifies the word V [55]. Attention predicts the current token in the decoder the same way as the seq2seq model's LSTM network by using the hidden state vectors from the encoder's predictions of previous words. However, self-attention uses all the words in the sequence regardless of the current position of the word as the attention operation is done at the same time to every token in the encoder. Multi-head attention refers to self-attention performed multiple times. This layer is an integral part of the encoder and decoder of a transformer model as it aims to obtain information from a diverse perspective in parallel by having several heads [11].

3) Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations (BERT) [12] is a language representation model based on transformer architecture. To compute a representation that reflects the context for each token, BERT is trained with two unsupervised sub-tasks to perform bi-directional prediction and sentence-level understanding. The tasks are 1) a masked language model (MLM), and 2) a next sentence prediction (NSP). For the MLM task, the model randomly masks 15% of the input sequence and is trained to predict the masked tokens using the context. NSP task trains the model to find the sequential relation between the provided pair of sentences by detecting when the second one follows the first one.

There are two BERT pre-trained models available depending on the size of the architecture.: BERT-base and BERT-large. The parameters of the BERT models are the number of Transformer blocks (L), the hidden size (H), and the number of self-attention heads (A). For the BERT-base model, the parameters

are L = 12, H = 768, and A = 12, with the total parameters of 110M and the BERT-large model has L = 24, H = 1024, A = 16, and the total parameters are 340M.

To train BERT, English Wikipedia (2,500M words) and BookCorpus (800M) [29] datasets were used. The BERT model was evaluated by applying it to the General Language Understanding Evaluation (GLUE) and Machine Reading Comprehension Task (SQuAD v1.1 and v2.0) for evaluation and achieved a new state-of-the-art performance for both tasks. BERT is designed to be a pre-trained model to be fine-tuned on task-specific data such as Question and Answering [56], Machine Translation [57], and Named Entity Recognition [58]. The procedures of both pre-training and fine-tuning for the BERT model are shown in the figure below.

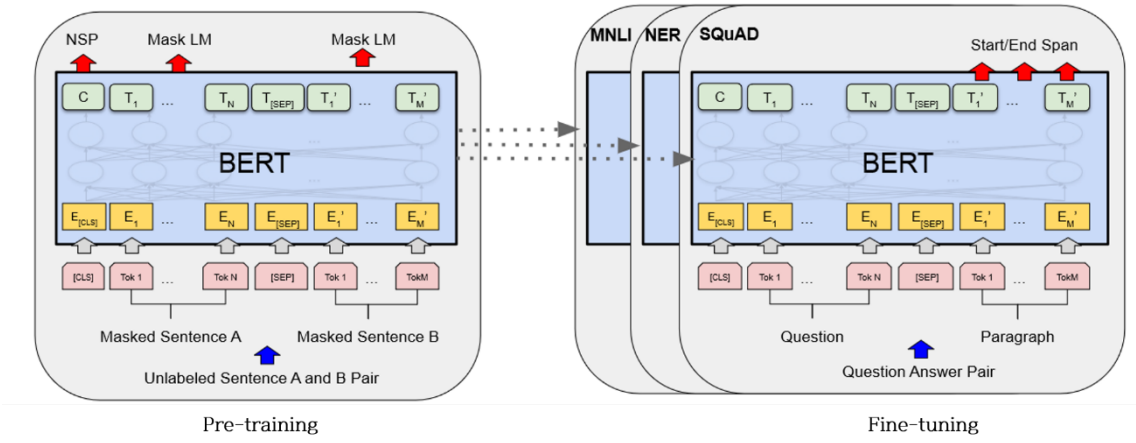


Figure 12 Pre-training and Fine-tuning Procedures of BERT [12]

4) Korean Language Models

With the advent of Transformer, comprehensive research and development of models specialized in the Korean language have been conducted in various fields including companies, schools, and individuals.

① Korean NLP Benchmarks

Various fine-tuned data and test datasets have been released to evaluate the

performance of diverse Korean natural language tasks. The NSMC[4] dataset, an emotional analysis dataset labeled in Naver movie review comment data, contains 200K reviews. The NaverNER[5] is a Korean NER dataset released in 2018 by Naver and Changwon University. To evaluate the NLU performance in the Korean language, KorNLI and KorSTS dataset was developed by KakaoBrain in 2020 [59]. The KorNLI dataset contains 942,854 training examples and 7,500 for evaluation. The KorSTS dataset comprises 5,749 examples automatically translated and 2,879 evaluation examples translated manually. In 2019, the Korean SQuAD dataset, KorQuAD[6], was released by LG CNS. It is a Korean Machine Reading Comprehension dataset consists a total of 100k+ pairs of questions and answers. Recently, the Korean version of GLUE (General Language Understanding Evaluation) [60], KLUE [61] was released.

② KoBERT

KoBERT[7] is a Korean pre-trained language model released by SKT-Brain. It follows most of the BERT's configuration with the tokenizer replaced as SentencePiece [8]from the WordPiece tokenizer. The model was trained using the Korean Wiki data that contains 5 million sentences and 54 million words.

③ KoELECTRA

KoELECTRA[9] is a language model based on the ELECTRA model. It is trained by determining whether the token generated by the discriminator is real or fake. The model is trained from 'Modu Corpus'[10] released by the National Institute of Korean Language, NamuWiki[11], and news data [62].

---

[4] https://github.com/e9t/nsmc

[5] https://github.com/naver/nlp-challenge

[6] https://korquad.github.io/

[7] https://github.com/SKTBrain/KoBERT

[8] https://github.com/google/sentencepiece

[9] https://github.com/monologg/KoELECTRA

[10] https://corpus.korean.go.kr/

[11] Large-scale Korean open domain encyclopedia.

④ KLUE-BERT

KLUE-BERT [12] model is a pre-trained language model that covers 8 downstream tasks. The model is pre-trained on the KLUE benchmark to help reproduce baseline models on KLUE.

[12] https://github.com/KLUE-benchmark/KLUE

3. Argumentation Mining

Argumentation is an intelligent discourse activity aimed at accepting or refuting proposed controversial claims or views [67]. Understanding argument structures in natural language provides a deeper insight into what is being said, thus by analyzing the argument structure and their premises and conclusions, we can comprehend both the content and the perspectives [17]. Argumentation can be found in various genres including court decisions, scientific texts, online forums, and debates [63]. Although several attempts have been made to analyze argument structures manually, with the overload of information, analyzing large volumes of text by hand has been proven to have limitations. In addition, while argumentation may take the form of formal propositions, in many areas of discourse, including law, reasonings are often based on informal arguments. These informal arguments often require further analysis to specify the structure of the argument since the relationship between the arguments is not explicitly expressed [64]. Given this problem, the concept of argument mining was introduced to ease the process of argumentation analysis. One of the first attempts at argument mining was made in 2007 which focused on mining argumentation from legal cases [51] [63].

Argumentation mining is the research area that regards natural language processing, argumentation theory, and information retrieval [64]. The task has been defined as "analyzing discourse on the pragmatics level and applying a certain argumentation theory to model and analyze the textual data" [22]. The goal of argument mining is to automatically detect arguments in documents including their structure and interaction between the propositions [64].

The process of argument mining reflects the human reasoning process in that arguments are first identified and their properties and relations are then detected to create an argument structure. Generally, an argument mining system takes unstructured text as input and produces a structured document as output where the arguments are detected and their relations are annotated to form an argument graph [14].

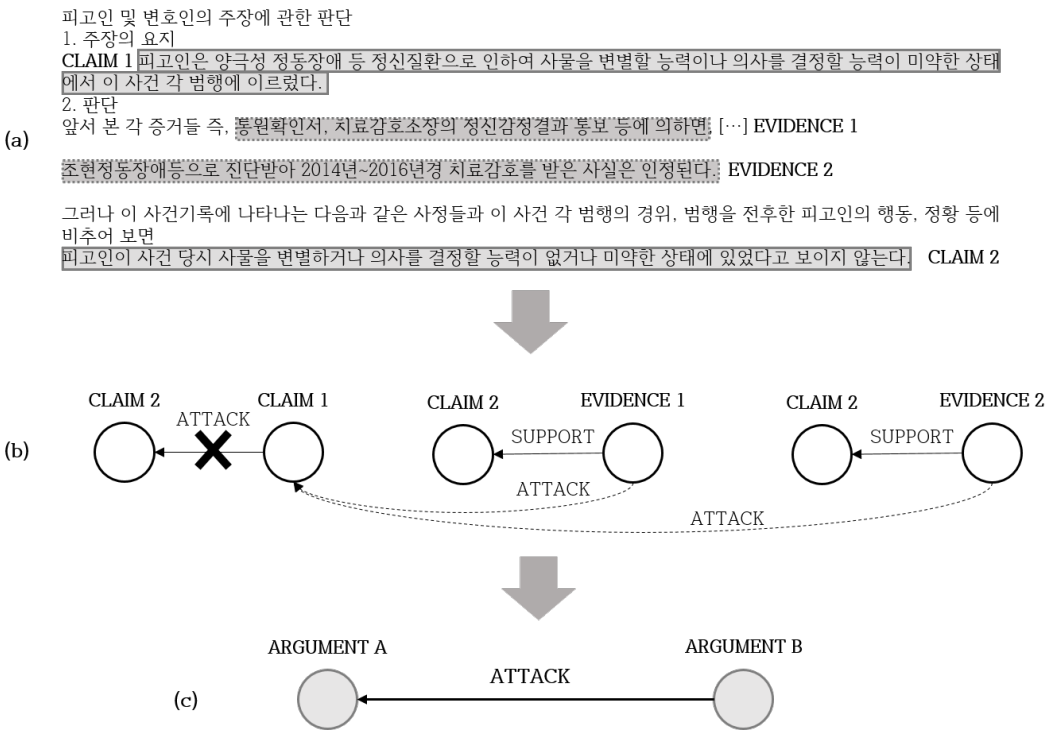


Figure 13 An Example of Argument Extraction

Figure 13 illustrates how to extract arguments from natural text automatically. First, argumentative sentences are recognized from the input document and their corresponding argument component. This process refers to Figure 11(a). Subsequently, the links between the argument components are predicted (Figure 11(b)), as well as the connection between argumentations (Figure 11(c)) to generate a complete argument graph [14].

According to Lippi and Torroni [14], there are two core tasks of argument mining: Argument Component Detection and Argument Structure Prediction.

1) Argument component detection is the first step in the argument mining system where argumentative sentences and their boundaries are identified from the general text. Different approaches have been suggested to address this task under supervised settings such as Support Vector Machines [64], [65], [66], Naïve Bayes [67], and Logistic Regression [68].

2) Predicting argument structure involves identifying the functional argument

components (e.g., premise, claim, etc.) and the connections between them. Generally, this task includes identifying the existence of relations (i.e., related or not related) and discovering the relation types (e.g., support or attack). In previous studies, the relation prediction has been tackled employing varying methods such as SVMs [23], Naïve Bayes [69], and Text Entailment [70].

While many successful approaches have been proposed for the task of argument component detection [51] [13], [71] [72] [73], predicting relations between argument components remains a significantly complex task and presents challenges to most machine learning methods as it requires high-level knowledge representation and reasoning [74].

Given the complexity of each task of the argument mining system which involves detecting, extracting, and predicting arguments and their relations from natural text, various techniques have been introduced for each step in the argument mining pipeline. The overall architecture of the argument mining pipeline is shown in Figure 14.

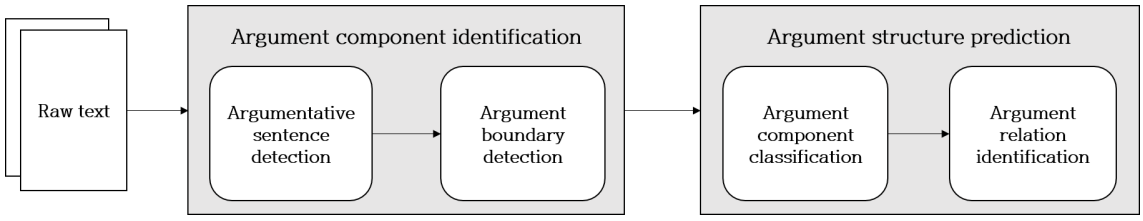


Figure 14 Argument Mining Pipeline

A. Argument Mining Methods

As mentioned in the previous section, the argument mining process involves two main tasks where the arguments are extracted and their relations are predicted. To achieve the aim of argument mining, the aforementioned tasks can be further split into a series of subtasks. In this section, we will break down the argument mining task into several individual challenges and look into the theoretical concepts and algorithms employed in related works.

1) Argument Component Identification

The first step in the argument mining pipeline is detecting argumentative portions from general text [3]. This task is defined as a text classification problem in most related works where argumentative parts of the text are recognized [51]. and is considered a crucial stage [75]. The task is generally approached by splitting it into two separate subtasks by first identifying the argumentative sentences and subsequently detecting their boundaries. However, in some works [23], [76] the second step is assumed to be previously detected by other means, thereby restricting the scope of the research to classifying the argumentive sentences only [14].

① Argumentative Sentence Detection

Detecting argumentative sentences is regarded as a classification problem and thus approached by choosing an appropriate classifier and features to identify argumentative texts from non-argumentative ones [3]. According to their work [14], this classification task can be approached using three options based on the argumentation model used to annotate the document. A binary classifier can be used to differentiate argumentative texts from non-argumentative ones. It is used when every sentence in the document can be annotated as argumentative or not. When the adopted argument model contains more than one argument component, a multi-class classifier is employed to discriminate all the components existing in the model. Lastly, a set of binary classifiers can be used when assuming that a sentence can contain more than one argument component. Hence, the classifiers are trained on individual components to predict their labels.

Most previous studies attempted to classify argumentative sentences by combining features extracted from the documents and classical machine learning classifiers such as SVMs [76], [64], [77], [23], Logistic Regression [78], [68], [79], Naïve Bayes [64], [80], and Random Forest [77], [23]. Among the classification algorithms that have been employed, SVM and Logistic Regression are the most frequently applied methods [74]. These classifiers are trained in supervised settings where both the text and their corresponding classes are provided during training, thereby generating models that predict labels on unseen texts [14].

② Argument Boundary Detection

The second stage of the argument mining system is the detection of argument

components' boundaries [23]. This task is defined as a segmentation problem where each argument component's starting and ending points are determined [81]. The boundary detection problem is dependent on the adopted argumentation model. In their study [81] using the claim/premise model, the average claim and premise sentences span 1.1. and 2.2. sentences respectively, whereas the IBM corpus considers the claim as a short part of the text that is always contained in a single sentence while the premise can span over multiple paragraphs [78], [68]. In some works, the task is disregarded assuming the sentences are already segmented [64], [15], [76].

The sentence segmentation problem can be approached by formulating the task as a sequence classification problem [82] and assigning a class to each word in the sentence thus distinguishing words within argument components. This method relies on the possibility of performing group classification where the sequential order of each word is considered. Using this framework has been proven to be robust for all types of relational data [83] by employing different methods such as Conditional Random Fields [77], or Recursive Neural Networks [84].

2) Argument Structure Prediction

The objective of this stage of the argument mining pipeline is to determine the connection between arguments. To detect the argument structure, the components that comprise the argumentation need to be identified first then subsequently their connections are determined. The related works generally separate the relation identification into two subtasks: identifying the connection (related or non-related) and determining the type of relation (e.g., support or attack). This is a challenging task as it involves high-level knowledge representation and inferential skills in order to understand the connections and relationships between argument propositions [74]. The retrieved relation information is used to construct argument graphs where the relations correspond to the edges.

① Argument Component Classification

In this stage, the goal is to determine the type of argument proposition (e.g., premise, claim, or conclusion). This task relies on the argument theory used to annotate the text such as Toulmin's argumentation model and determines the sentences into the applied argument propositions. An additional class of non-argumentative can be used as a part of the argument component identification

classifier. Different classifiers have been used to achieve this task. In their study [85], a two-step argument mining pipeline was used by first classifying the sentences as claims and further distinguishing them into support, oppose or propose types. Mochales-Palau [13] and Stab and Gurevych [23] employed SVMs for classifying premises and claims and partial tree kernels were used in Lippi and Torroni's work [86] for the same task.

② Argument Relation Identification

After recognizing the argument components, the links between each component are predicted. This task is influenced by the underlying argument model, for instance when dealing with a simple claim/premise argumentation model, the structure can be formalized as a bipartite graph [14]. For a more sophisticated argument model such as the Toulmin model containing six components, the task becomes more complex since components can be left implicit.

Several approaches have been made to extract argument pairs using different methods. Mochales and Moens [64] have introduced a manually built context-free grammar to predict relations between argument components using the grammar rules that follow the typical patterns found in legal texts. In Stab and Gurevych's work, an approach to predict links in the claim/premise argument model was proposed using binary SVM classifiers [23]. Cabrio and Villata explored a method using text entailment aiming to infer the relationship between given argument pairs such as support or attack [70].

3) Argument Mining Using Transformer Architectures

Recently, several works have utilized transformer-based models for automatically identifying and extracting argumentative components and detecting the existing relations among them. The contextual word embeddings in the form of BERT and ELMo were used in Reimers et al.'s work to improve the performance of the argument mining task [87]. In their work [88], a BERT-base model was proposed for argument component classification along with relation detection in a persuasive online discussion corpus. Lastly, Ruiz-Dolz et al. [89] conducted an exhaustive analysis of the behavior of the transformer-based models for argument mining tasks. They obtained a macro F1-score of 0.70 with the US2016 debate corpus and a 0.61 score with the cross-domain corpus proving the model's effectiveness in various domains of the corpus.

A. Argument Mining Corpus

For an argumentation mining system to successfully work, creating a properly annotated dataset is essential. The performance of the model generally depends on the quality and amount of the corpus. However, constructing an annotated argument corpus is considered an expensive and time-consuming task especially for domain-specific datasets, as identifying components and their relations is challenging even for humans [81]. Therefore, a great amount of effort and resources are devoted to the development of consistent annotations. Furthermore, a corpus built with a specific goal or domain is challenging to directly apply to a general argument mining pipeline. Thus, using a dataset well-subjected to the purpose of the model is crucial. An overview of the different corpora used in argument mining tasks is shown in Table 2.

1) Argument mining in literature

The corpus for argument mining has been collected in various fields such as education, online content, newspapers, medicine, and law.

In the field of education, argument mining is applied to persuasive essays as they contain logical perspectives on the given topic. Stab and Gurevych identified the argument components and the structures using an annotated corpus of persuasive essays[13] [73]. The same dataset was used in Eger et al.'s work [90] which proposed an argument mining system using neural networks. They found that the detecting argument components and relations should be addressed separately but jointly modeled.

For online-based content, Wikipedia articles have been used to create a debate corpus. The most well-known dataset using this corpus is IBM's project debater datasets [14]that allow several argument mining tasks. This corpus aims to collect context-dependent arguments and premises related to a given subject [14]. Using this corpus, Levy et al. [68] attempted to automatically detect context-dependent claims given topics from debates. Another well-annotated corpus based on user-generated content was developed by Habernal et al. [81]. They aimed to model arguments following a variant of the Toulmin model which contains 990 English comments to articles and forum posts. Additionally, datasets

[13] https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2421

[14] https://research.ibm.com/haifa/dept/vst/debating_data.shtml

were collected from resources such as online reviews [91], blogs [22], and newspapers [92].

Recently, argumentation mining has also been attempted in the field of medicine and healthcare. The datasets created from this domain aim to build ontologies that explain the correlations between symptoms and diseases or assist healthcare professionals to develop treatment plans based on the provided evidence. In Stylianou and Vlahavas's work [93], they aimed to identify related evidence in medical literature for the practitioners to make choices based on the given information. To achieve this goal, they created an argument mining pipeline using the Transformer architecture. Similarly, Mayer et al. also employed the Transformer model [94] in classifying argument components and predicting the relations from medical trial abstracts.

2) Argument mining in the legal domain

In the legal domain, argument mining has been applied to recognize the premises, claims, and argumentation structures in court decisions or legal cases to facilitate the process of identifying similarities and differences between cases [74]. For instance, Mochales and Moens [64] proposed a work on the European Court of Human Rights that attempts to detect argument components and structures. The ECHR texts are easy to exploit for the task of argument mining as they contain a standard type of reasoning and structure of argumentation [16]. In their study, argument components were identified using features such as n-gram, verbs, punctuations, and argumentative patterns. For the structure extraction, a context-free grammar was constructed to parse the text, achieving a 60% accuracy. This study implies that argument mining in the legal domain is capable and led to the following work by Teruel et al. [95] where a new corpus of ECHR containing annotations with premises and claims along with the support and attack relationships.

Table 2 An Overview of Argument Mining Datasets

Domain	Document Source	Size	Task	
			Component Detection	Relation Identification
Education	Persuasive essays [73]	402 essays	O	O
Web	Wikipedia [68]	32 debate motion	O	
	Comments on articles, and forum posts [81]	990 English comments	O	
	Newspaper [96]	100 editorials	O	
Medicine	MEDLINE [94]	6.8k Randomized Controlled Trial abstracts	O	O
	EBM-NLP corpus [93]	5k abstracts of medical publications	O	O
Legal	ECHR judgments [64]	7 judgments	O	O
	ECHR judgments [95]	47 judgments	O	O

III. Text annotation and dataset

One of the challenges of argument mining is the lack of a properly annotated corpus, including argument components and their relations [97] [98]. As seen from the previous chapter (see section Argument Theory), various approaches have been suggested to the definition of argument and the structure of argumentation, thus there is no unified structure in building the argument corpus. Thus far, a few corpora that annotated arguments have been suggested. Araucaria [97] is one of the most well-known corpora for argument mining tasks that include information on argumentive relations. The corpus consists of arguments from various genres, including newspapers, parliamentary records, judicial reports, and online discussion boards. Several researchers have used Araucaria for different tasks, e.g. [64], [99], [51]. The most comprehensive collection of annotated arguments is AIFdb17[15] [100]. It is a publicly accessible database containing more than 14,000 AIF (Argument Interchange Format) argument maps and includes more than 1.6 million words and 160,000 claims in 14 different languages.

For legal argument mining purposes, the ECHR corpus is often used [101] [102]. It consists of 42 legal decisions from the European Court of Human Rights (ECHR) with three types of clauses annotated: premise, conclusions, and non-argument parts of the text and their relations. Two lawyers were hired to annotate the document based on a guideline. Then a third lawyer was selected to analyze the annotation and explain the discrepancies. From this process, a new guideline was created which was given to the fourth annotator and obtained 80% inter-rater agreement using Cohen's kappa coefficient [101].

Considering the objectivity of this study which aims to build a system that helps the Korean investigators who handle legal case documents written in the Korean language, none of the currently existing corpora was suitable for the task. Therefore, in this study, a new corpus that comprises Korean legal case documents annotating argument components and their argumentative relations is

[15] https://www.aifdb.org/search

generated.

1. Corpus creation process

A. Data collection process

The Online Access to Court Record system (판결서 인터넷 열람 서비스) [16]operated by the Korean court allows users to access criminal court decisions electronically for cases confirmed from January 1st, 2013 by entering the sentencing date, court name, case number, and related laws as search terms. The judgments are provided as an image file (PDF) that cannot be comprehended by machines[17]. Before this change, only Supreme Court decisions were openly accessible, which posed a limitation as a source of data. While the newly changed system significantly improved the accessibility to criminal cases, there is still a limitation as a fee of 1,000 KRW is charged per case, and can be viewed and downloaded only within 24 hours after the initial reading [103].

Then the collected judgments in image PDFs are converted into structured text files through Optical Character Recognition (OCR) for a machine to comprehend. The OCR process outputs a text file from an image file through preprocessing, text detection, and text recognition. In this study, the judgments in image PDFs are converted into text files using an OCR program produced using the NAVER CLOVA OCR API, a paid service optimized for the Korean language.

To select data that have similarities to crime investigation reports, the following criteria were prepared.

1) Lower court judgments addressing disputes over facts as an issue

2) Judgments addressing homicide (Article 250 of the Criminal Code) and rape (Article 297 of the Criminal Code) an issue

---

[16] https://www.scourt.go.kr/portal/information/finalruling/peruse/peruse_status.jsp

[17] According to the revised 「Regulations on the Publication of Judgment Through Electronic Mail」 (전자우편 등을 통한 판결문제공에 관한 예규), civil and criminal judgments posted as of July 5th , 2021 will be provided as a file capable of text search.

3) Judgments containing the Defendant's claim and the Judge's evaluation

Among the judgments that meet the above criteria, 84 cases of judgments that resulted in an acquittal of the defendant or granted electronic monitoring to defendants were dismissed. As a result, a total of 224 cases of homicide and 32 cases of rape were used to build the corpus.

B. Argumentation structure in criminal judgments

Although there is no specific regulation on the details and the order of the items, the criminal judgments are written following a format that generally can be structured as below.

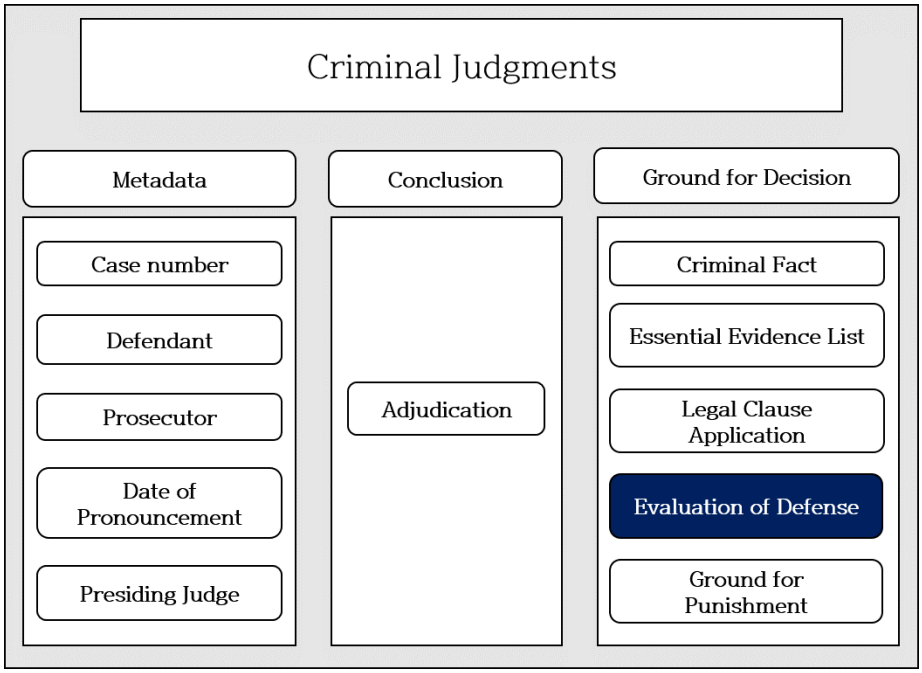


Figure 15 Criminal Judgments Structure

Criminal judgments can largely be divided into 3 parts: Metadata, Conclusion, and Ground for Decision. The metadata contains basic information about the case, i.e. court name, case number, and personal information of the related parties to the case. Adjudication corresponds to the final verdict of the case usually written in a one-line sentence [2]. Ground for Decision provides detailed information on the verdict in the order of criminal fact, essential evidence list, legal clause

application, evaluation of defense, and grounds for the punishment. The 'evaluation of defense' contains the argumentation process of the court by either accepting or denying the defense's claim. Therefore, in this study, we focus on the arguments found in the 'Evaluation of Defense' section.

C. Data pre-processing

As seen from the section above, the judgment documents can be separated into multiple sections. However, they are generally unstructured, thus the data is preprocessed before annotation to increase the quality of the corpus.

1) Text segmentation

The text segmentation task is the extraction of text fragments that constitute a document's argument structure [16]. Text segmentation can be regarded as identifying the elementary argumentative units and various hypotheses have been proposed for the criteria that include these units, e.g. clauses [101] and sentences [73]. However, Korean legal documents contain multiple conjunctions and phrases in one sentence [104], thus applying these segmentation approaches can be challenging.

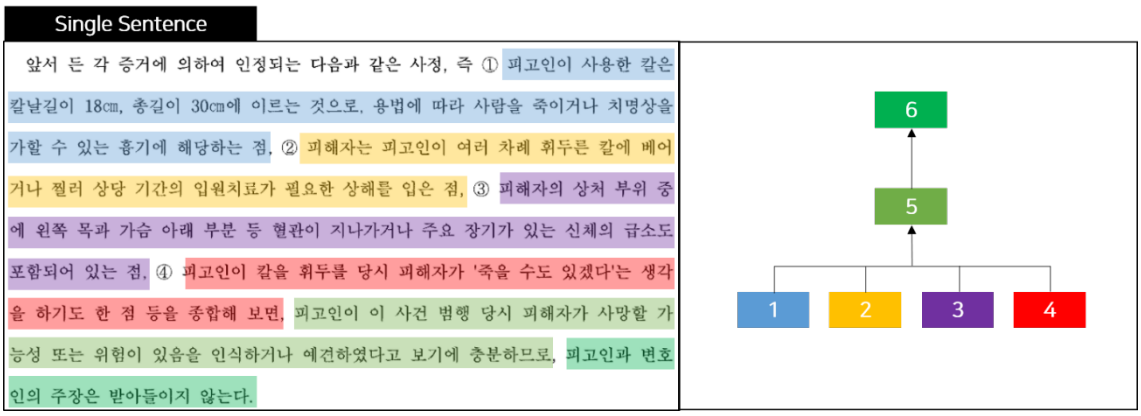


Figure 16 An Example of a Complexed Argument Found in Judgments

Figure 16 is an example of sentences from a homicide case that demonstrates the complexity of legal documents. The sentence can be divided into multiple premises and a conclusion, thus the sentence can have multiple labels. This problem is also identified in Poudyal's study which aimed to differentiate the components by using a set of conjunctions (e.g. that, because, and) and

punctuations as keywords from the ECHR corpus [37]. Assigning multiple labels to a sentence goes against the purpose of understanding the structure of arguments by automatically analyzing the components [2]. Since the text segmentation problem is heavily influenced by the argument model adopted [14], we attempted to split the documents based on the characteristics of the argument structure used in this study. Therefore, several phrase segmentation rules were created to suit the annotation scheme applied in this study.

① Punctuations

On our dataset, punctuation is a good indicator when detecting phrases. However, there are cases where (1) commas are used to list items or (2) periods are used to mark abbreviations or separate dates. Therefore, the above cases were defined as exceptions, and phrases were not divided if applicable. The examples for phrase segmentation and exception are shown with a ₩n as an indicator for the beginning of a new line.

- *위와 같은 행위로 피해자가 사망할 가능성 또는 위험이 있음을 인식하거나 예견하였다고 할 것이므로, ₩n 피고인에게 살인의 범의가 있었음이 충분히 인정된다.*
- *exception (1) 위와 같은 상처의 모양, 깊이 등에 비추어 볼 때*
- *exception (2) 대법원2006. 4. 14. 선고 2006도734 판결 등 참조*

② Quoted passages

Criminal judgments often use quote testimonies from witnesses or various sources related to the case. Phrases that fall under this category are wrapped with quotation marks or parentheses. The quoted phrases are treated as a single component and split by the end of the quotation marks or punctuations within the quotation. Below is an example.

- *"2008.4.말경 어느 날 피고인은 1시간 정도 늦게 집에 들어왔는데 별말이 없고 표정이 어두웠으며 평상시와 달랐다." ₩n "피고인은 집에서 혼자 술을 마셨고 갑자기 중국에 가서 살자고 하였다."*

③ Inference phrases

The inference phrase includes keywords such as “비추어 보면”, “종합하여 보면” or “의하면” which translates to “based on” or “in light of” in English. This type of phrase is split as it can be separated into two different components (i.g.,

premise and claim). An example is shown below.

- *위 자창의 위치, 깊이, 길이에 비추어 볼 때 ₩n 피고인이 범행 당시 주저하지 않고 매우 강하게 피해자의 목 부위를 회칼로 찔렀음을 알 수 있다.*

④ Correlative conjunction

Correlative conjunctions are used to emphasize and connect two words or phrases simultaneously, parallelism being the primary goal. In our dataset, "뿐만 아니라", which translates to "not only", is used frequently to indicate the correlations between two phrases within a sentence. Thus, a phrase that contains this conjunction is split as shown below.

- *위와 같은 증인의 진술 내용뿐만 아니라 ₩n 증인의 모습이나 태도,말투,표정,진술의 뉘앙스 등에 비추어 전반적으로 신빙성이 있다.*

⑤ Subordinating conjunctions

Subordinating conjunctions link independent clauses to dependent clauses. By doing this, the subordinating conjunction demonstrates the relationship between the phrases, which is often a cause-and-effect relationship or a contrast. Phrases including conjunctions such as "이므로" (because) or "때문에"(since) are split as shown below.

- *따라서 피고인은 이 사건 범행 당시 사물을 변별할 능력이나 의사를 결정할 능력이 없는 심신상실 상태에 있었으므로 ₩n 피고인에 대한 이 사건 공소사실은 모두 무죄이다.*

⑥ Nominal Phrases

A nominal phrase performs the same grammatical function as a noun. This is a commonly used phrase in judgments that ends with "한 점" or "한 사실" which can be translated as "the fact that" in English. It is used to describe a fact either given or found by the court.

- 피고인이 이례적으로 아들을 조퇴시키고 갑자기 아산으로 놀러 간 점, ₩n 피고인이 아산에 가기 전에 일을 그만두고 그 뒤로 일하지 않았으며 아산에 다녀온 후에는 중국으로 떠날 준비만 서두른 점,

Following this phrase segmentation guideline, the annotators manually split the data, and annotation of each phrase was undertaken.

D. Annotation

1) Toulmin+ model

The pre-processed corpus is then annotated following the argumentation scheme built on the concept of Toulmin's argumentation model, as it best represented the characteristics of arguments found in legal documents.

The modified version of Toulmin's argumentation model is referred to as the Toulmin+ model. The Toulmin+ model expands and re-constructs the original Toulmin model by changing the components. The components removed in our adaptation are Qualifiers that refer to modal verbs and Rebuttals, which specify the conditions for defeating the claim [67, p. 92, p.94]. In our newly built argument model, the added components are Inference, Expert Opinion, and Issue Conclusion, and the relations between the components are expressed using relational components such as attack or support which replace the Rebuttal. The table below describes the argument components of our model.

Table 3 Toulmin+ Components

Types	Argument Component	Description
Facts/ Evidence	Datum (D)	Evidence or data supporting the claim
	Expert Opinion (EO)	An opinion given by an expert Includes testimonies or documents by experts
	Backing (B)	A reference to precedents or legislation
Hypothesis/ Conclusions	Warrant (W)	A logical bridge between a datum and a claim Generally approved rules or principles, such as precedents, and common knowledge.
	Inference (I)	A sub-claim of a sub-issue A deduction from a datum, and a logical bridge between a datum and a claim
	Claim (C)	The main argument by the judge and the defense Must be identified first as the other components' relations are classified

		based on claims
	Issue Conclusion (IC)	A conclusion of an issue

The application of the Toulmin+ model on our dataset is shown in Table 4.

Table 4 An Example of Toulmin+ Model Applied on our Data

Argument Component	Phrase
Datum (D)	피해자의 가슴 자상의 크기는 4cm 정도로써 복장 뼈 바로 앞까지 상처가 나 있는 점,
Inference (I)	피해자의 진술은 일관성이 있고 구체적이어서 믿을 수 있으며,
Backing (B)	(대법원 2009.11.26. 선고 2009도 7918 판결 등 참조),
Warrant (W)	살인죄에서 살인의 범의는 반드시 살해의 목적이나 계획적인 살해의 의도가 있어야인정되는 것은 아니고,
Expert Opinion (EO)	국립과학수사연구원의 감정결과에 따르면 피고인이 사용한 총기는 당시 발사거리(피해자 I의 경우 2.9m. 피해자 H의 경우 2m)에서 살상의 위력이 충분한 점,
Claim (C)	피고인이 이 사건 범행 당시 정신적 장애로 인하여 사물을 변별할 능력이나 의사를 결정할 능력이 미약한 상태에 있었다고는 보이지 않으므로,
Issue Conclusion (IC)	피고인 및 변호인의 이 부분 주장은 받아들이지 않는다.

2) Annotation process

Using the Toulmin+ argumentation model, the annotators annotated the judgments following a guideline including the annotation process. The process takes three steps:

1. Topic identification: We ask the annotators to read the entire text before starting with the annotation task to identify different topics within the document. This is an essential process for our data as a legal judgment may contain more than one issue in the same case (i.e., 1. intent to kill and 2. history of mental disorder). This also can contribute to improving the inter-annotator agreement [23].
2. Argument component annotation: Annotators label the argumentative phrases using the Toulmin+ model. Assigning more than one component to a phrase is not allowed.

3. Linking components with argumentative relations: The annotators identify the structure of arguments by linking two components with their argumentative relationships (e.g., support, attack, or parallel). This process will reveal the entire structure of the document by annotators marking the defeated element.

A detailed description of each criterion listed in the annotation guideline is shown in the table below.

Table 5 Toulmin+ annotation process criteria

Types	Description	
Toulmin Num	Definition	Number of the Toulmin argument structure to which the component belongs
	Format	[Integer]
	Rule	Can't assign more than one label
	Example	1
		The component belongs to the first Toulmin structure.
Component	Definition	Argument components in Toulmin+ model
	Format	[Toulmin_num]_[Component=string]_[Component_num=int]
	Rule	Can't assign more than one label
	Example	1_w_1
		The component is the first warrant belonging to the first Toulmi structure
Relation	Definition	A component that has a relationship with another component. The left-hand side is the child component, which the right-hand side supports/attacks/in parallel with.
	Format	[Toulmin_num]_[Component=string]_[Component_num=int]
	Rule	The relation of each component is only defined once.
	Example	1_w_1 → 1_c_1
		The first warrant in the first Toulmin structure has a relationship with the first claim of the same structure.
Relation type	Definition	The relation types between the two components can be support, attack, or parallel.

	Format	[s, a, p]
	Rule	The relation type corresponds to the number of related components
	Example	1_w_1 → 1_c_1 (Relation type: S)
		The relation type between the two components is support.
Defeated	Definition	The status of a component being defeated by another component
	Format	[yes, no, na]
	Rule	The defeated element corresponds to the number of related components.
	Example	1_i_1 → 1_c_1 (relation type = S) 1_i_2 → 1_i_1 (relation type = A) 1_i_3 → 1_i_2 (relation type = A) ∴ 1_i_2 → defeated
Phrase	Definition	The component's corresponding phrase.

The fully annotated data can then be exported into JSON (JavaScript Object Notation) files and as input data for our machine-learning models. A sample JSON data is shown below.

```
{
        "meta":

        {
                "case_id": 4,
                "title": "부산지방법원2020.7.10선고2019고합649,2019고합6",
                "type": "살인"
                },

        "annotation_data": [

        {
                "toulmin_No.": 1,
                "component": "1_D_1",
                "relation": "1_C_1",
                "relation_type": "S",
                "role": "na",
                "defeated": "yes",
                "phrase": "피고인이 피해자 I(이하 이 항에서는 '피해자'라고 한다)의
                    허벅지를 칼로 1회 찔러 상해를 가한 사실은 있으나,"
                },

        {
                "toulmin_No.": 1,
                "component": "1_C_1",
                "relation": "na",
                "relation_type": "na",
                "role": "d",
                "defeated": "yes",
                "phrase": "피해자를 살해하려는 고의가 없었다."
                },

        ]
}
```


Figure 17 Sample CSV data of an annotated data

3) Argument Visualization

When visualizing the argument structure as a graph using the Toulmin+ model, the components correspond to the node, and the relations refer to the edges that link the components. The nodes in the graph are placed in their respective layers, namely, the Evidential layer (E-layer) and the Inferential layer (I-layer). Components in the E-layer correspond to the information or data in the court decision including datums, backings, and expert opinions. Nodes in the I-layer represent the inferential phrases related to argumentative statements in our data that correspond to inferences, warrants, claims, and issue claims.

The Toulmin+ model imposes restrictions on edge types based on the layers. The nodes within the same layer can support, attack, and be in parallel

relationships with each other except for datum and backing. However, for nodes placed in two different layers, they can only form support or attack relationship. Hence, edges from an E-layer to an I-layer represent the inference rule by providing information as input for the inference.

An illustration of the visualized Toulmin+ model is shown in Figure 18.

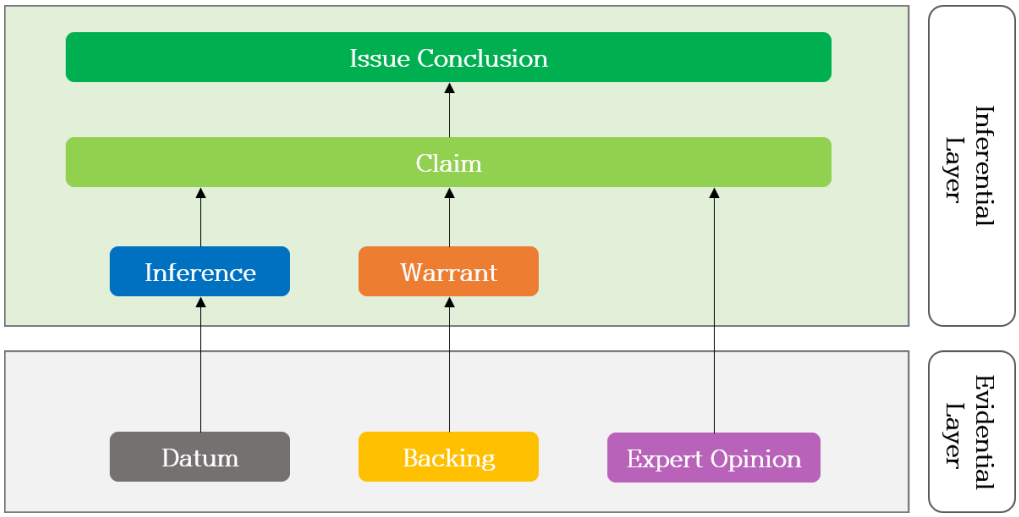


Figure 18 A Visualized Toulmin+ Model

According to this process, the formalized visualization patterns of the Toulmin+ argument structure can be classified into a total of 13 types. This demonstrates that the court decisions follow a certain process of logic, therefore argument structures that deviate from this pattern may be considered to be less logical. The retrieved argument patterns are shown in Figure 19.

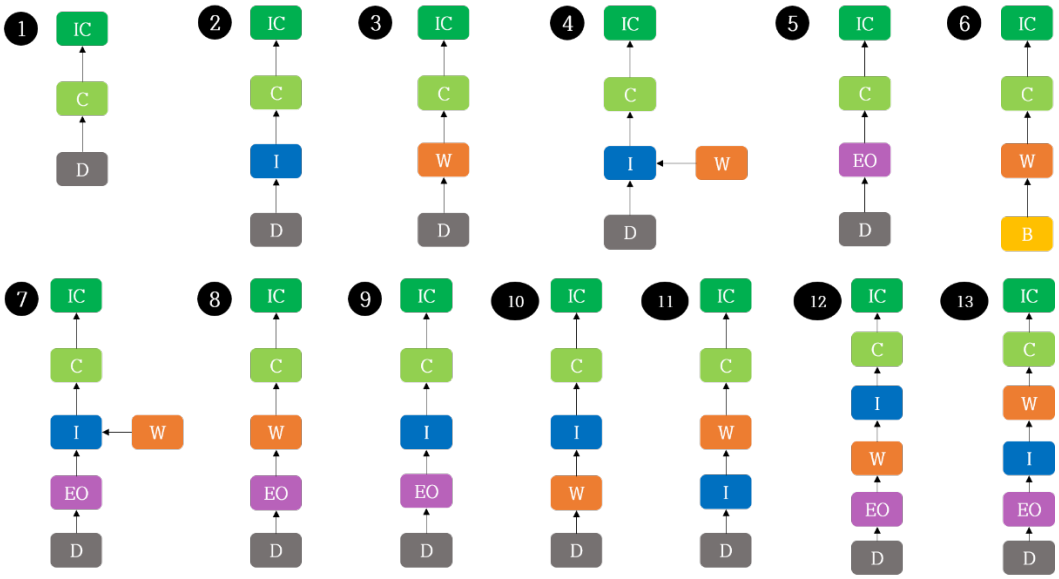


Figure 19 Toulmin+ Visualization Patterns

2. Corpus evaluation

A. Inter-rater Reliability

To build an objective and generalized dataset, an inter-rater reliability (IRR) evaluation of the annotated data was conducted. It is necessary to verify that raters agree on the analysis criteria to obtain reliable text data [105]. According to [106], the IRR evaluation is effective in identifying raters' consistency and the consensus of raters' perspectives of components.

Several studies conducted IRR evaluation on argument annotated datasets using various methods to calculate IRR [101], [73], [76], [94]. The IRR evaluation methods frequently used in the literature regarding text data are described below.

Table 6 IRR evaluation methods

IRR evaluation method	Description
Fleiss' Kappa [107]	Extends Cohen's Kappa by not limiting the number of raters.
Krippendorff's U-alpha [105]	Measures the IRR for any number of evaluators from continuum data including text and video. The degree of concordance is calculated using the entire data.

For our research, the annotators who are legal informatics graduate students were paired into two, and the annotated data were submitted for an IRR agreement evaluation using two different methods: Fleiss' κ and Krippendorff's α [105]. The overall process of the IRR evaluation is shown in Figure 20.

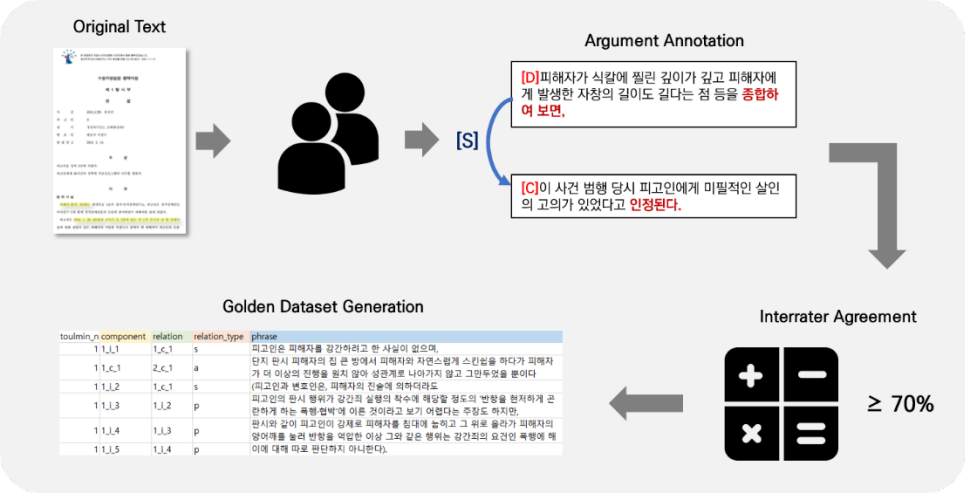


Figure 20 IRR Process

We consider the number of each phrase and evaluate the presence of the argument component tagged for the individual phrase. The discrepancy between the two annotators was discussed and refined the annotation guideline based on this discussion. Sample analysis of the annotation discrepancy is shown in the table below.

Table 7 An Example of Annotation Discrepancy

Discrepancy type	Case name	Line number	Found discrepancy	Rater A	Rater B
Inference – Datum	B60	50	Phrase	피고인은 피해자를 사망하게 할 수 있다는 가능성 또는 위험을 인식하거나 예견한 상태에서 차량 밖에서 피해자의 목을 졸라 피해자로 하여금 의식을 잃게 한 사실을 인정할 수 있으므로,	
			Component	Inference	Datum
			Description	The phrase is tagged as an inference focusing on the described opinion of the judge.	The phrase is tagged as datum focusing on the described evidence.
Conclusion	The phrase is a judge's inference drawn from the defendant's behavior, therefore it should be annotated as an inference.				

The mean IRR score of Fleiss' κ was 0.782 and Krippendorff's α is 0.784. Considering that scores over 0.7 are indications of a good agreement [15], we can interpret that annotators have reached a consensus on the meaning of the Toulmin+ components.

B. Corpus statistics

From 256 court judgments, a total of 12,911 argumentative phrases were retrieved and used to create our corpus. Within the corpus, an imbalance between labels is noticed, with datum being the most dominant component. This is representative of court decisions, as datums are frequently used as inputs for argumentative statements made. The table below shows the total number of each component in our corpus.

Table 8 A Statistics of Corpus

Components	Count (phrase)	Ratio
Datum	4,530	0.35
Claim	1,537	0.11
Inference	4,263	0.33
Warrant	1,262	0.09
Backing	326	0.02
Issue Claim	460	0.03
Expert Opinion	533	0.04
Total Count	12,911	1

IV. Research design

1. Proposed architecture

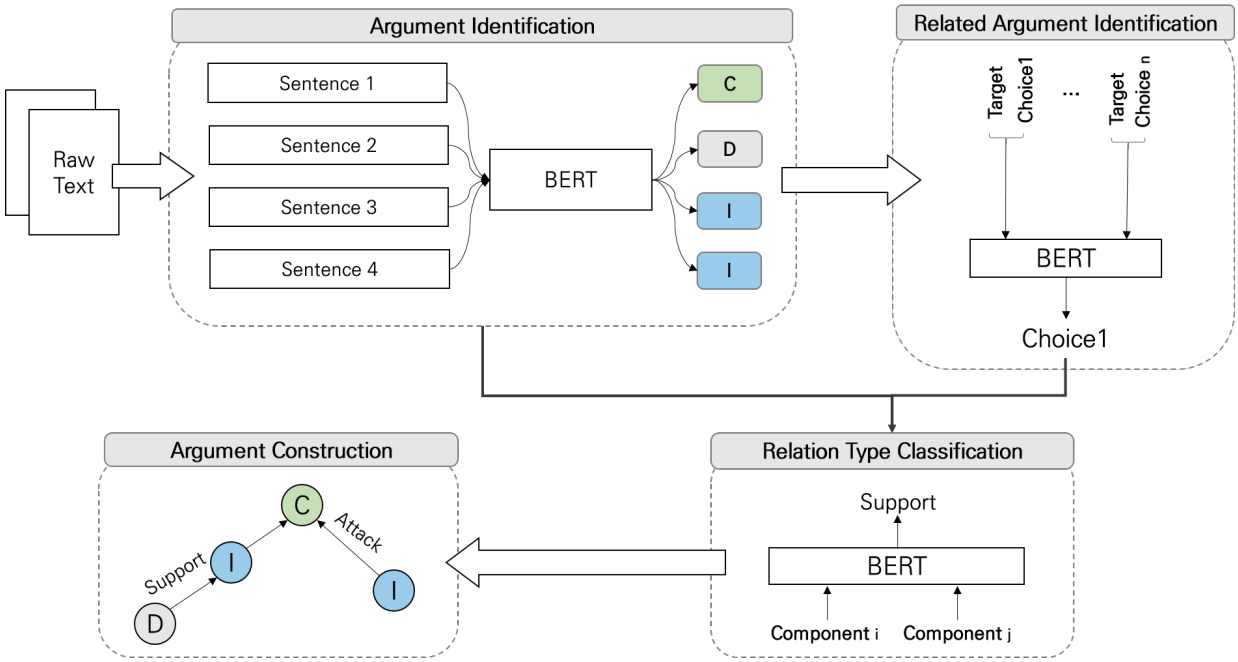


Figure 21 Overview of the Proposed Model Architecture

This chapter provides an overview of the proposed model. The model aims to identify argument structures in a legal corpus using Transformer-based language models. To the best of our knowledge, this is the first work addressing the legal argument mining problem using a Transformer-based model. The architecture of the proposed model has multiple modules, as shown in Figure 21. The modules work sequentially by first identifying the argument components, detecting the related phrase pairs and their relation types, and finally extracting an argument structure and visualizing it as a graph.

The following sections describe the detailed algorithms and models used in each process.

2. Argument component classification

This is the first step in the proposed model architecture with its aim to correctly classify the elements of the Toulmin+ model labeled for each phrase in the corpus. As described above, most of the argument mining works approach this task assuming the boundaries of argument components as given, thus we use segmented phrases as input for our model.

To achieve this goal, a pre-trained BERT model to classify the components. Figure 22 describes the overview of the task.

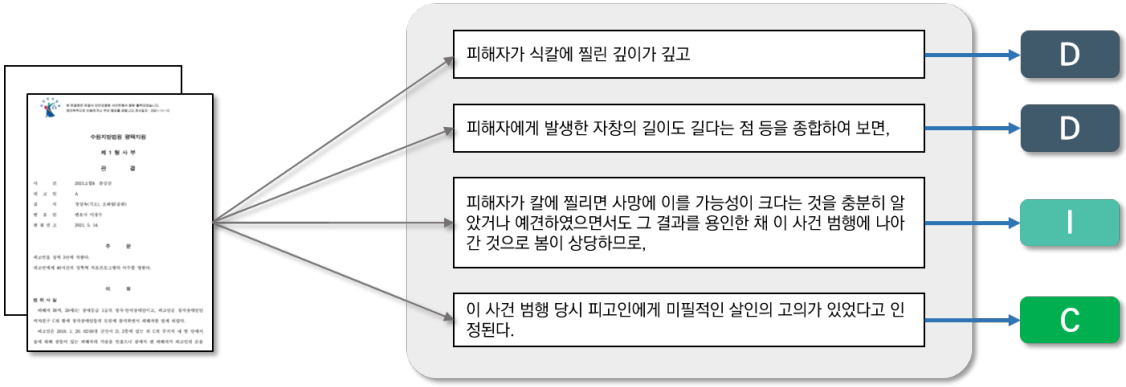


Figure 22 Overview of the argument component classifier

A. Multi-class classification using BERT

The problem addressed in this part of the study can be defined as the multi-class classification problem. We approach this by fine-tuning a pre-trained BERT model to fit the multi-class classification task for our domain-specific dataset. In our implementation of the BERT model, we use the pre-trained Korean BERT model (KoBERT[18]) developed by SKTBrain. The KoBERT model is trained on the

[18] https://github.com/SKTBrain/ KoBERT

Korean language using the Korean Wikipedia and Korean news data each containing more than 5 million sentences and 20 million sentences. Using such pre-trained models, semantically richer representations can be created from the input sequence modeled at a token level by training on a large dataset [93]. The KoBERT model has a vocabulary size of 8,002 and provides a SentencePiece tokenizer trained specifically for the tokenization task of the Korean language.

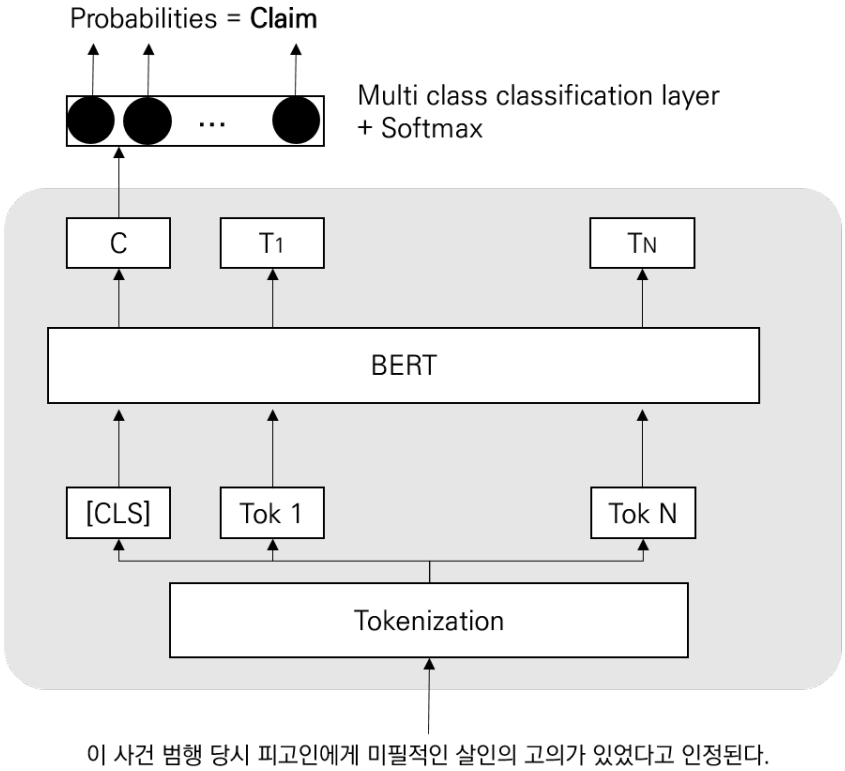


Figure 23 Proposed Architecture of BERT-based Argument Component Classifier

The architecture for our proposed classification model is shown in Figure 23. Here, the [CLS] is a special token indicating the beginning of all input sequences, and Tok 1 to Tok N refers to every word in the input sequence that has been tokenized using a tokenizer trained on Korean Wiki and news text. The embedding vector of the input sequence is derived from multiple layers of Transformers. Using the pooled output vector of the pre-trained model which corresponds to

the [CLS] token in Figure 16 as an input.

To prevent the model from overfitting, the model consists of a dropout layer and a fully connected layer. A dropout rate of 0.5 was used in this study. The model also uses weight decay as a regularization method which adds a penalty to the loss function to have smaller weights to prevent overfitting. The 768-dimensional embedding vector corresponding to the output of the [CLS] token outputs a total of 8-dimensional vectors through the dropout layer and the fully connected layer, and each of the features of this vector represents the probability of belonging to a specific argumentative component.

3. Argument Relation Identification

After detecting the argument components, we detect relationships between the different components. Legal documents such as court decisions consist of argument groups that create relations with each other. The related arguments are grouped and referenced by another argument group. Recognizing the argument relation is much more challenging than identifying argument components as it requires understanding the connections and relational properties of the arguments.

This part of the study aims to identify the argumentative relations from the document by defining the task as a sequence classification problem and using transformer-based neural architectures. Recent works regarding the argument relation mining problems address this using Recurrent Neural Network-based methods (e.g. LSTMs, BiLSTMs, etc.) [71], [108], [109]. However, the Transformer architecture can improve the RNN model by allowing the model to capture a longer range of dependencies within a longer input sequence using multiple attention modules [89]. Considering the nature of our corpus, a long and complex input sequence is expected. Therefore, we use transformer-based models that have attention modules for identifying the relational properties between argument components. In their work [94], Mayer et al. suggested a transformer-based approach for classifying argumentative relationships from texts by predicting possible link candidates for each component and then classifying relationships only for plausible pairs.

Referring to this method, the task for our model is divided into two interrelated procedures: (1) identifying the related argument pairs from the text, (2) and classifying their corresponding relation types. Figure 24 illustrates the overview of this process.

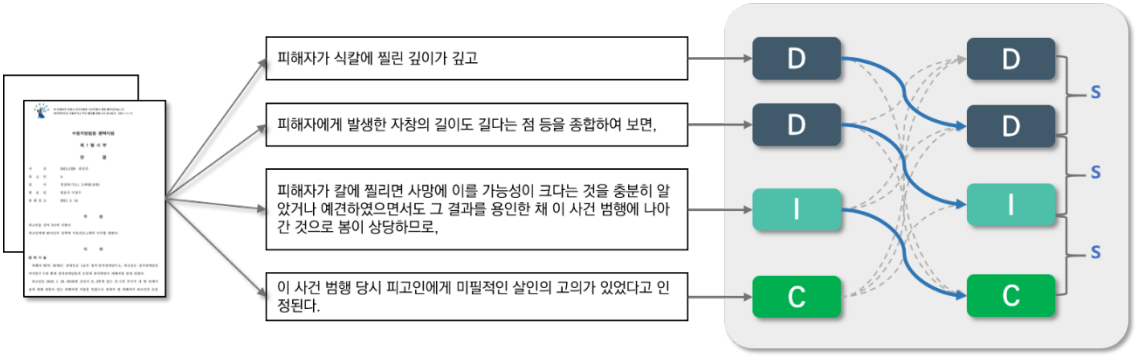


Figure 24 Overview of argument relation identification

A. Relation candidate classification

The first task for our relation classification model is to identify the possible argument link pairs. We create a multiple-choice setting to retrieve such pairs, where argumentatively possible links are predicted by considering the other combinations. In this approach, each component (i.e., target component) is given a list of all other components as possible relationship candidates and determines which component is most likely to be related to the target component among the candidates. We believe giving a target and several candidate options to the model will improve the model's reasoning skills significantly compare to training the sentence-to-sentence relationships since the model learns the contextual relationships from the given choices. In their work [110], the multiple-choice approach has been defined as grounded commonsense inference. A similar approach was proposed by Mayer et al. [94] where the relation classification problem was tackled by creating multiple-choice settings.

For our multiple-choice model, we use DistilKoBERT[19], a smaller distillation

[19] https://github.com/monologg/DistilKoBERT

of the pre-trained KoBERT model. It follows the general architecture as BERT with *token-type embedding* and *pooler* removed [111].

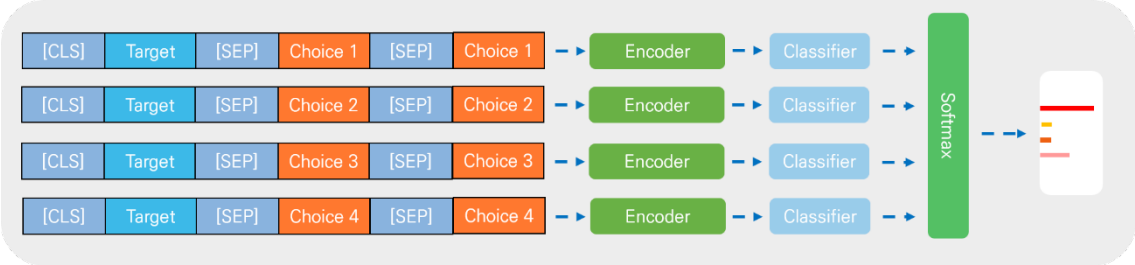


Figure 25 Proposed Architecture of BERT-based Multiple-Choice Classification Model

The proposed model is trained to select the correct answer from four choices. The architecture for our proposed multiple-choice model is illustrated in Figure 25 In this model, the target component and one of the choices from the relationship candidates are concatenated into a sequence. Afterward, each sequence is encoded to be represented by a vector which is passed into the classifier creating a logit vector for all choices. The vectors are then transformed into the probability vector through a softmax layer. The choice with the highest logit value is considered to have a link with the target component.

For the experiment, a total of 7,545 multi-choice sequences were created. Figure 26 shows an example of the dataset for our model. Here, a target phrase and four choices are given as candidates with label 0 indicating that choice 1 is the correctly related phrase to the target.

Target	Choice 1	Choice 2	Choice 3	Choice 4	Label
그러나 그로 인하여 피고인이 사물을 변별할 능력이나 의사를 결정할 능력이 미약한 상태에까지 이르렀다고 볼 만한 자료는 찾아보기 어렵다.	피고인이 음주 또는 정신질환으로 인하여 사물을 변별하거나 의사를 결정할 능력이 없었다거나 미약한 상태에 있었다고는 보이지 않는다.	피고인이 이 사건 범행 당시 심신미약의 상태에 있었다고 주장하므로	이 법원이 적법하게 채택한 증거들에 의하면,	피고인이 식칼로 피해자의 배를 향해 찔렀고,	0

Figure 26 An Example of a Multi-Choice Dataset

B. Argument relation type classification

After detecting the related argument pairs, their corresponding relation types are identified. Predicting the relations between arguments is an extremely challenging task as it involves high-level knowledge representation and inferential issues to understand the connection and relationships between the arguments [74]. The arguments may support and attack one another, and the retrieved argument relationships are used to construct argument graphs.

The relation classification task can be approached using different methods including some of the classical machine learning solutions such as SVM, Naïve Bayes, and Textual Entailment [23], [69], [70]. However, in our study, we use a transformer-based model by treating it as a sequence classification problem assigning the most probable class to two text inputs. Transformer-based models have been achieving state-of-the-art performance for tasks involving the classification of text sequences [12]. Thus, in our study, a Natural Language Inference (NLI) [112], [113] based approach is used to tackle the sequence classification problem. NLI aims to infer the relationship between the hypothesis sentence and the premise sentence. Given a premise, the model is asked to determine whether a hypothesis is true (entailment), false (contradiction), or undetermined (neutral). An example of an NLI dataset is shown in Table 9. The NLI task is expected to achieve general goals in Natural Language Understanding (NLU) [112], [114], such as learning sentence representations [115] and evaluating NLP models [116]. Therefore, in our attempt, we formulate NLI as an argument sequence classification task where the model is asked to predict whether the relationship is support, attack, or parallel given a pair of argument phrases. The architecture of our NLI-based relationship classifier is shown in Figure 27.

Table 9 An example of an NLI dataset[20]

Premise	Hypothesis	Label
A soccer game with multiple males playing.	Some men are playing a sport.	entailment
A man inspects the uniform of a figure in some East Asian	The man is sleeping	contradiction

[20] The Stanford Natural Language Inference (SNLI) Corpus (https://nlp.stanford.edu/projects/snli/)

country.		
An older and younger man smiling.	Two men are smiling and laughing at the cats playing on the floor.	neutral

For our argument sequence classification model, we use a pre-trained BERT model on the Korean language [61], KLUE[21], and fine-tuned it to fit the relation detection task. By fine-tuning the model's parameter to fit the relation detection task, the method's modified task is the following: given input statements A and B, what percentage P is the chance that A and B belong to Support, Attack, and Parallel.

The model takes two phases as inputs which are passed through the BERT network to get the embeddings u and v. Once the vectors are generated, we concatenate the concatenation (u,v) and absolute element-wise difference (|u-v|) into a long vector to extract relations between u and v. This vector is then passed to a softmax classifier, which predicts our three classes (support, attack, parallel). Thus, we aim to demonstrate that phrase encoders trained on natural language inference can learn sentence representations that capture useful features.

[21] https://github.com/KLUE-benchmark/KLUE

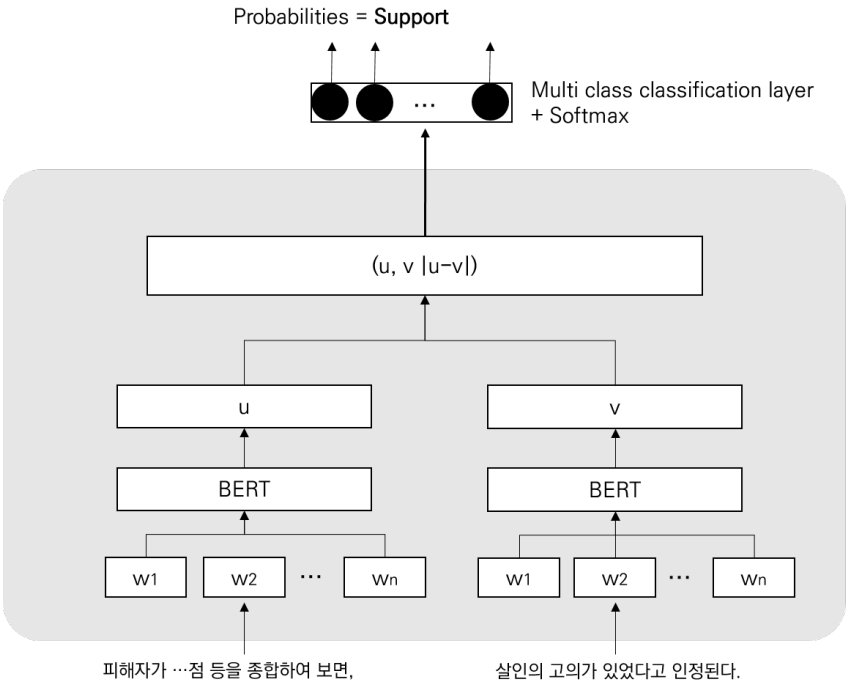


Figure 27 Proposed Architecture of NLI-based Relation Type Classification Model

For the experiment, a total of 14,055 argument phrase pairs were created and used as a corpus. Within the corpus, an imbalance in the attack label was noticed. This is because multi-labeling for relation types was not allowed in our annotation scheme. In the case of the court decisions, while the Defendant's claim is concisely expressed in one or two sentences, the rest of the document that attacks the Defendant's claim is expressed in several sentences. Thus, expressing the one-to-one attack relationship in our annotation process that only allows single-labeling is challenging.

The final corpus for our experiment was then created containing the four argument relationships used in the Toulmin+ argumentation model. An example of our corpus is provided in Table 10 along with the statistics on each relation type in our corpus in Table 11.

Table 10 An Example of Relation Type Dataset

Phrase 1	Phrase 2	Relation type
피고인이 위 범행 당시 술에 취하여 사물을 변별할 능력이나 의사를결정할 능력이 미약한 상태에 있었다고 보이지 아니하므로,	따라서 피고인 및 변호인의 위 주장 역시 받아들일 수 없다.	Support
살인의 고의가 없었다는 취지로 주장한다.	이 사건 범행 당시 피고인에게 미필적인 살인의 고의가 있었다고 인정된다.	Attack
범행 이후 피고인이 자전거를 타고서 여기 저기 다니거나 전화를 거는 등	이 사건 범행에서 나타나는 범행의 수단 및 방법,	Parallel
그 정도가 사회적 상당성을 결여한 경우를 가리키는 것이므로,	피고인이 이 사건 범행 당시 상당한 양의 술을 마신 사실은 인정된다.	No-relation

Table 11 A Statistics on Relation Types of our Corpus

Component	Count	Percentage (%)
Support	5,186	36
Attack	859	6
Parallel	5,344	38
No Relation	1,421	10
Total Count	12,810	100

4. Argument structure extraction

Based on the classification models proposed in the previous sections, an argument structure representation method is devised. Argument representation is one of the most productive research trends, especially in the legal domain [117], [118], that aims to analyze and evaluate complex argumentation through visualization. Several resources and tools for argument visualization are available including Araucaria [99], Carneades [117], and AIFdb [100]. However, these tools require users to manually identify the components and their relations to construct an argument graph. Therefore, in our study, we present an argument

structure extraction method that aims to automatically construct an argument graph based on the Toulmin+ argument model by integrating the two models created in previous stages of our study; the Argument Component Classifier and Argument Relation Identifier, and visualize their structure based on the information.

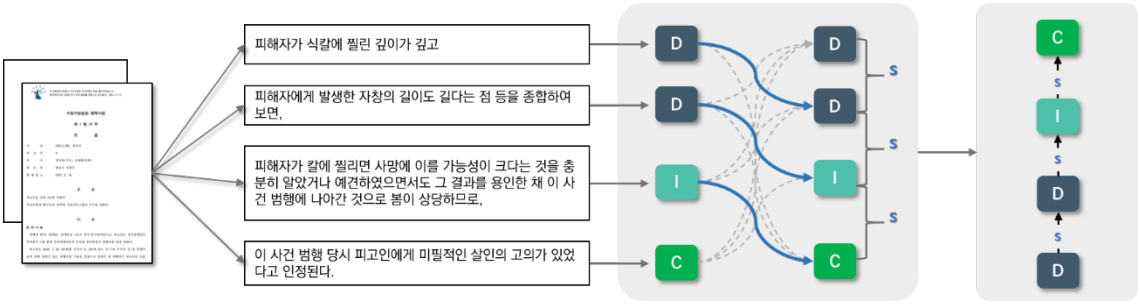


Figure 28 Overview of the Argument Structure Extraction System

The proposed argument structure extraction system deals with the identification of the internal structure of the arguments i.e. identification of argument components as well as classification of their relationships. Therefore, the extracted argument structure can be used to analyze the logical completeness of the argumentation. An overview of the argument structure extraction module is presented in Figure 28.

For visualizing the argument structures as graphs, a python package NetworkX [119] library was used. NetworkX is an open-source network analysis tool providing data structures for representing graphs. In our experiment, the graph module regards every component as nodes and relations as edges that connect the components. The root nodes are always set as Issue Conclusion, however, in case the component does not exist, Claim becomes the root node.

The goal is to represent the argumentative phrases in a graph, thus extracting the internal argument structure of the document. The experiment results are explained in Section 5.3.

5. Summary

This chapter describes the proposed architecture for a system that identifies argument structures from texts in the legal domain. First, a transformer-based approach to identify the argument components from the legal documents was proposed. Second, a Multiple-choice and NLI-based approach is applied to detect the argumentatively related phrase pairs and the relationships they hold. The method is challenging as it requires high knowledge representation skills. Finally, an approach aiming to extract the argument structure and visually represent it is investigated. In this step, the previously defined argument components and their relationships are used as nodes and edges to construct an argument graph.

V. Results and discussion

This chapter evaluates the proposed systems in Chapter Ⅳ and discusses the results of the experiments. The experiments are tested in a Google Colab[22] environment using the Python 3 Google Compute Engine backend with 12.72 GB RAM.

For the multiclassification tasks, the metrics used in the measurements are the macro f1 score, precision, and recall. 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 (P) is the correct information among the identified instances calculated by dividing the count of true positives (TP) by the sum of true positives and false positives ($TP + FP$) $P = \frac{TP}{TP+FP}$. Recall (R) scores are the divided score of true positives (TP) and the sum of true positives and false negatives ($P + FN$) $R = \frac{TP}{TP+FN}$.

1. Argument Component Classification

In this section, we discuss the results of the proposed argument component identification process. From the entire corpus of 12,911 phrases, 20% of the data randomly extracted were used to evaluate the performance of the trained classification model. Thus, a total of 7,685 phrases were used for training, 2,562 phrases for validation, and the remaining 2,561 phrases were used for testing the model's performance. Table 12 summarises the distribution of our dataset.

---

[22] Colaboratory, or "Colab" for short, is a product from Google Research. Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs (https://colab.research.google.com/)

Table 12 Distribution of the Components

Component	Train set	Validation Set	Test Set
Datum	2,671	932	927
Inference	2,309	839	832
Warrant	758	239	265
Backing	185	65	76
Claim	661	322	272
Issue Conclusion	282	90	88
Expert Opinion	333	91	109
Undefined	486	164	169

The experiment was conducted using pytorch_kobert_model with 12 attention heads, a hidden size of 768, 12 transformer blocks, with 7 labels to classify. Batch size was set to 16 with the maximum sequence length of 256 input tokens after identifying the input sequence distribution. The model was trained with a learning rate of 5e-5 with Adam optimizer for 15 epochs.

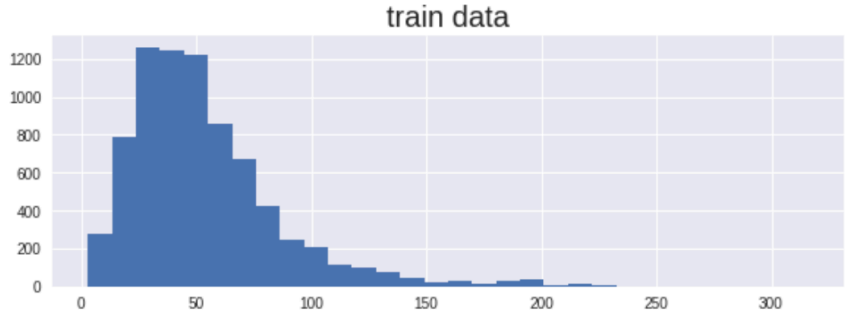


Figure 29 Input Sequence Length Distribution for Component Classification Dataset

A. Evaluation of the Classifier

Table 13 contains a comparison of our proposed model with the baseline model (SVM) [2] tested in the Korean court decision dataset.

Table 13 Evaluation of the Component Classification Models

	KoBERT	Support Vector Machine [2]
F1	0.9244	0.7466
Precision	0.9281	0.7082
Recall	0.9209	0.9329

The results show that the proposed transformer-based model improves performance in all metrics than our baseline model. From the results, it can be inferred that using the Transformer architecture, especially the encoders can increase the performance.

The table below shows the f1 score (macro) calculated on each label classified by our model. Overall, most components were classified correctly by achieving an average score of 90% on every metric. More specifically, the classification of issue conclusion, warrant, and backing showed the highest accuracy. This can be attributed to the fact that the phrases labeled as the above three are relatively short in length, and have certain words that appear frequently, These characteristics result in high classification accuracy.

Table 14 Performance Evaluation of the Toulmin+ Argument Component Classification

Components	Precision	Recall	F1-score
Issue Conclusion	0.9886	0.9886	0.9886
Claim	0.9162	0.8913	0.9036
Inference	0.8758	0.8723	0.8741
Warrant	0.9807	0.9585	0.9695
Backing	1.0000	1.0000	1.0000
Expert Opinion	0.8252	0.7798	0.8019
Datum	0.9024	0.9180	0.9102
Undefined	0.9360	0.9583	0.9471

In the next section, we provide a detailed investigation of the misclassified data to improve the model's performance.

1) Analysis of Misclassified Data

The figure below shows the distribution of each component's actual label and the predicted value. Our model classified most of the components successfully, however, some notable misclassifications arose in predicting certain components.

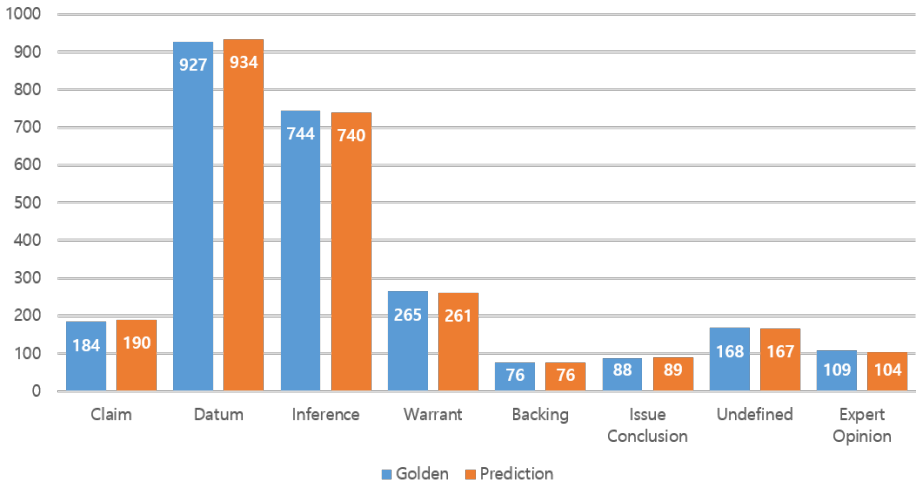


Figure 30 Model Prediciton Comparison Plot

One of the common mistakes for the component classifier is misclassifying components as Inference, especially in the case of Claims, Warrants, and Datums that share similar phrasal patterns found in Inference. As an Inference corresponds to a hypothesis drawn from the evidence, it contains words that show a subjective view of the speaker. Some of the commonly found words in Inferences are '인정된다' (accepted), "하지만"(but), or "힘껏"(powerfully). The predicted results for the three components that are frequently misclassified as Inference are presented in figure 30.

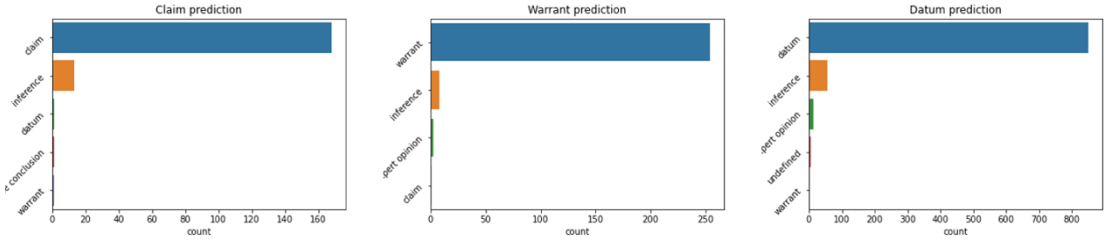


Figure 31 Predicted Results for Claim, Issue Warrant, and Datum

Correctly classifying these components is a challenging task even for human annotators as they require inferential knowledge. The misclassification examples are shown below.

Table 15 Examples of Shared Inference Patterns Predicted by the Model

Phrase	Golden Label	Predicted Label
이 사건 범행 당시 피고인에게는 적어도 살인의 미필적 고의가 있었음이 충분히 인정된다.	Claim	Inference
일반인의 입장에서 피고인에게 하반신 마비를 치료할 능력이 있다고 믿을 수 있을지 다소 의문의 여지는 있을 수 있으나,	Warrant	Inference
4 피고인은 피해자에게 '죽여버린다'고 소리치며 들고 있던 칼로 힘껏 찔렀던 점,	Datum	Inference

Another mistake made by the model is the misclassification of Inference and Expert Opinion phrases as Datum which is shown in Figure 31.

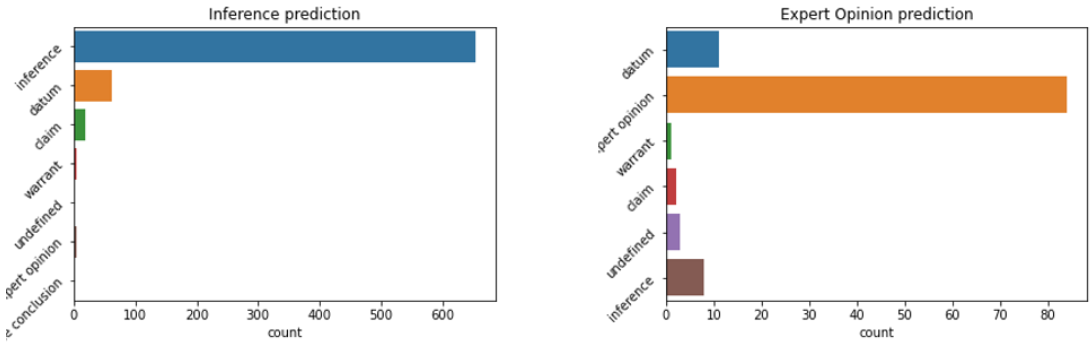


Figure 32 Predicted Results for Inference and Expert Opinion

Datums are used to express various factual and evidential information, hence the length of the phrase and the range of the vocabulary used are diverse. These characteristics of datum are reflected in the misclassification of inference and expert opinion. From our observation, inference phrases that are misclassified as datums take the form of cited testimony usually containing the word "취지" (intent). In our annotation framework, we treat phrases containing such words as Inferences as they imply the speaker's opinion rather than hard facts. Our

analysis also showed that expert opinions misclassified as datum were mainly expert witness statements which can be confused as evidential information. The misclassification examples of Inference and Expert Opinion are given in the table below.

Table 16 Examples of Shared Inference Patterns Predicted by the Model

Phrase	Golden Label	Predicted Label
당시에는 그 동안에 쌓였던 감정이 순간적으로 폭발을 해서 눈이 뒤집혀 낫을 휘두르기 시작할 때부터 큰 아들이 말릴 때까지 제정신이 아니었다."는 취지로 진술하고 있고,	Inference	Datum
④증인 전00은 피해자가 이전에 자궁 등 산부인과적 수술을 하여 우측 장간막과 장이 유착된 상태였기 때문에	Expert Opinion	Datum

Our model showed robustness in classifying Backing, Issue Conclusions, and Undefined components. The predicted results for each component are shown in the figure below.

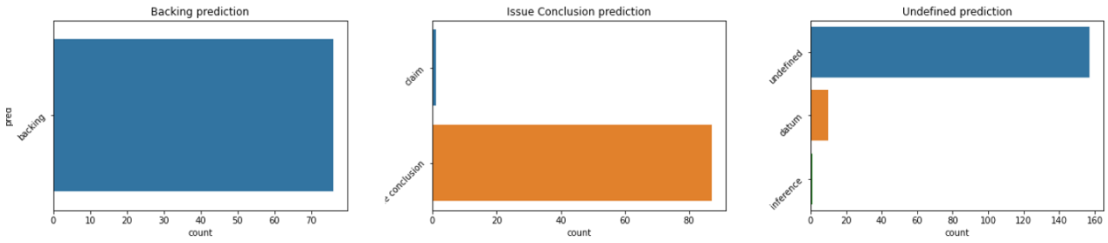


Figure 33 Predicted Results for Backing, Issue Conclusion, and Undefined

The model's robustness can be understood due to the linguistic patterns in our training data which the model has exploited. The aforementioned components share a similarity in that they have certain words used frequently which the model learns to use as a shortcut in prediction. Some examples of these patterns are described in the table below.

Table 17 Examples of Linguistic Patterns Predicted by the Model

Phrase	Golden Label	Predicted Label
(대법원 2000. 8. 18. 선고 2000도 2231 판결 등 참조).	Backing	Backing
피고인 및 변호인의 이 부분 주장은 받아들이지 아니 한다.	Issue Conclusion	Issue Conclusion
판시 증거들에 의하여 인정되는 다음과 같은 사정 즉,	Undefined	Undefined

The analysis of the model's performance indicates that our fine-tuned model can be directly applied to the legal documents to identify each argument component. It also suggests that the proposed Toulmin+ model is effective in identifying the argument components.

2. Argument Relation Identification

In this section, the results of the proposed argument relation identification process are given. According to Lippi and Torroni, the goal of this task can be defined as predicting the connection between the input texts [14].

Here, we divided the task into two separate experiments to identify the related argument pairs and then classify the relationships they hold. The first experiment trained a BERT-based multiple-choice model to predict the correctly related phrase from possible relation candidates. For the second experiment, a BERT-based NLI model was used to classify the relationships between argument pairs. Hence, the models for the two experiments are each called the Multi-Choice classifier and NLI classifier.

A. Evaluation of the Multi-Choice Classifier

For the first experiment, the monologg/distilkobert model[23] was used to fine-tune our model with a multiple-choice classification layer. The model has 12 attention heads, a hidden size of 768, and 3 layers and the task is to select the correct label from the four candidate choices, given a target phrase. From the corpus of 6,732 phrase sequences, 60% of the data was used to train the model and the remaining 40% was evenly split into validation and test sets to evaluate the model's performance. A detailed distribution of our dataset is shown in Table 18.

Table 18 Distribution of the Labels

Label	Train	Validation	Test
Targets	4,019	1,356	1,357
Choice 0	1,237	410	404
Choice 1	789	263	282
Choice 2	1,216	419	406
Choice 3	776	263	265

The Multi-Choice model was trained with Cross Entropy loss with a learning rate of 2e-5 for 20 epochs. We used the batch size of 16 with the maximum sequence length of 256 input tokens by considering the input sequence distribution of the train data which can be seen in Figure 34. The table below provides our model's performance calculated on each label. The model showed an average score of 70% on every metric indicating that the distilkobert model can deliver a reliable result on our dataset even though the model is trained on general data. Based on such results, we can infer that the Wiki data used to train KoBERT includes vocabulary used in court decisions. Furthermore, considering that the multi-choice model presented in [94] scored 66% in the relation classification task, the improved f1-score of our model suggests the model's ability to infer the relationship between a given target phrase and candidate phrases.

[23] https://huggingface.co/monologg/distilkobert/blob/main/config.json

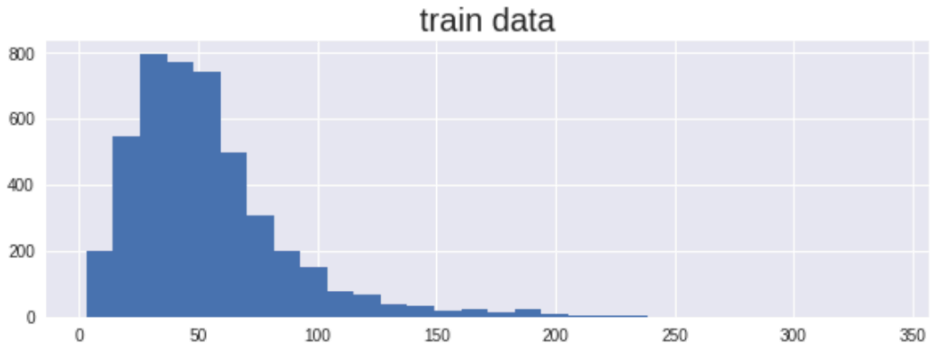


Figure 34 Input Sequence Length Distribution for Multi-Choice Dataset

Table 19 Evaluation of the Multi-Choice Model

Label	Precision	Recall	F1-score
Choice 0	0.80	0.73	0.76
Choice 1	0.73	0.77	0.75
Choice 2	0.83	0.78	0.81
Choice 3	0.64	0.76	0.69
Macro average	0.75	0.76	0.75

1) Analysis of Misclassified Relations

In an effort to evaluate the model more thoroughly, we analyze the errors made by our Multi-Choice classification model. From the analysis, we observed that this misclassification is largely due to the omission of contextual information. For argumentation, it is very common to simplify the concepts used in the past by not explicitly mentioning them or replacing them with pronouns [89]. Thus, this lack of contextual information complicates the automatic identification of argument relations. To better understand this problem, we give the following example predictions of our model on the multiple-choice dataset in Table 20.

In the first example, our model misclassified the second choice as a related phrase to the target. In fact, by reading the phrase in Choice 2, it may be considered that an argumentative relationship exists with the target phrase. However, it can be inferred that the error was caused due to the model's failure to recognize the pronoun in the target sentence (*"이로 인하여"*) refers to the victim's wound in the Choice 1 phrase (*"피해자가 입은 상처"*). The second and

third examples also reveal similar misclassification patterns by failing to capture the semantic relationship between the target phrase and the gold label phrase. Here, the phrase segments in bold indicate the existence of relations. In these situations, a possible solution to avoid errors is to provide additional information about the mentioned pronouns.

Table 20 Example Predictions of Multi-Choice Model

Example 1	
Target: **이로 인하여** 상당한 양의 출혈이 발생하였다.	
Gold label	Choice 1: 이처럼 **피해자가 입은 상처**의 깊이와 길이가 상당하고,
Predicted	Choice 2: 따라서 피고인에게는 피해자를 살해할 고의가 없었다.
None	Choice 3: 살인미수죄의 고의는 반드시 살해의 목적이나 계획적인 살해의 의도가 있어야 인정되는 것은 아니고,
None	Choice 4: 자기의 행위로 인하여 타인의 사망이라는 결과를 발생시킬 만한 가능성 또는 위험이 있음을 인식하거나 예견하면 족한 것이며,
Example 2	
Target: **심신미약의 상태**에 있었다고 판단된다.	
Gold label	Choice 1: 따라서 **이에 반하는** 피고인과 변호인의 주장은 이유 없다.
None	Choice 2: 자기의 행위로 인하여 타인의 사망의 결과를 발생시킬 만한 위험이 있음을 예견· 용인하면 족하며 그 주관적 예견 등은 확정적인 것은 물론 불확정적인 것이더라도 미필적 고의로서 살인의 범의가 인정될 수 있다
None	Choice 3: (대법원 2011. 12. 22. 선고 2011도 12927 판결).
Predicted	Choice 4: 2) 위와 같은 법리에 비추어 본다.
Example 3	
Target: 피고인 역시 **이러한 점**을 인식할 수 있었다고 보인다.	
Gold label	Choice 1: **피고인이 승용차를 계속 진행할 경우 피해자들이 바퀴에 깔리거나 역과될 가능성**이 있었고,
None	Choice 2: 살인의 고의는 없었고,
Predicted	Choice 3: 피해자들의 머리나 얼굴 등 급소를 향해 망치를 휘두르거나 내리친 적도 없다.
None	Choice 4: 살인죄에서 살인의 범의는 반드시 살해의 목적이나 계획적인 살해의 의도가 있어야인정되는 것은 아니고,

B. Evaluation of the Relation Type Classifier

After recognizing the argumentative pair from the text, the second part of the argument relation identification experiment was conducted to classify the relationship types between the given phrase pairs. For this purpose, we fine-tuned the klue/bert-base model for the NLI task with 4 labels using the TFBertforSequenceClassification[24] model. The entire corpus of 12,810 phrase sequences was split into the train, validation, and test sets, with ratios set at 60%, 20%, and 20%. The data distribution is shown in Table 21.

Table 21 Data Distribution of the NLI Dataset

Label	Train	Validation	Test
Support	3,142	1,032	1,012
Attack	520	180	159
Parallel	3,174	1,076	1,094
No relation	850	274	297

We trained the NLI model with the cross-entropy loss using the Adam optimizer with a learning rate of 5e-5 for 5 epochs and set the batch size to 16 and the maximum sequence length to 256 considering the input sequence distribution of the train data shown in Figure 35. The model's performance on each label is provided in Table 22. The results show that our model achieved an average of 91% f1-score proving that the NLI-based approach can provide reliable results in capturing the argumentative relationships. These results surpass the performance of previous work on relation identification that achieved a 0.751 macro F1 score using SVM [73], thus showing how the Transformer-based architecture can improve the performance in detecting argumentative relations. When comparing the classification performance on each label, the model showed a good performance on most labels, however, the performance dropped to 80% when classifying the attack label. From this result, we can infer

[24] TFBertForSequenceClassification is a Bert Model transformer with a sequence classification head on top. A detailed description of the model can be found on hugging face. (https://huggingface.co/transformers/v3.0.2/model_doc/bert.html#tfbertforsequenceclassification)

that the class imbalance within our dataset has affected the model to degrade.

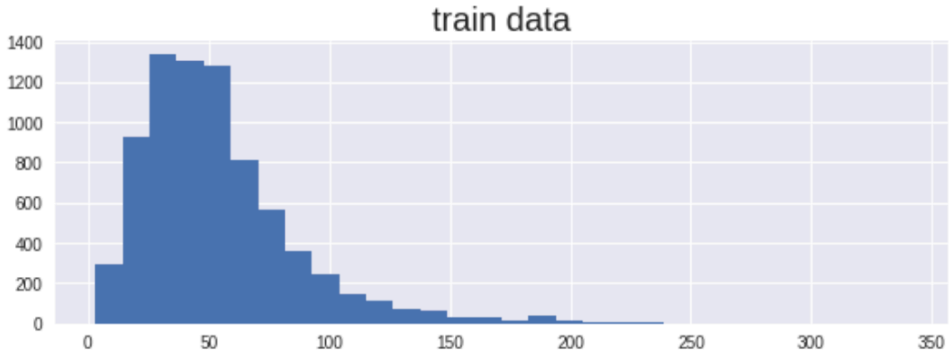


Figure 35 Input Sequence Length Distribution for NLI Dataset

Table 22 Model Performance on the NLI Task

Label	Precision	Recall	F1-score
Attack	0.8000	0.8054	0.8027
No-relation	0.9959	1.0000	0.9979
Parallel	0.9182	0.9525	0.9351
Support	0.9459	0.9061	0.9256
Macro average	0.9160	0.9153	0.9150

1) Analysis of Misclassified Data

In order to analyze the misclassification of the NLI model, we investigate the predicted values in our test data.

Table 23 Comparison between the Model Prediction and Golden set

		predictions			
		Support	Attack	Parallel	None
golden	Support	**934**	9	68	1
	Attack	29	**119**	11	0
	Parallel	54	5	**1035**	0
	None	0	5	0	**297**

Table 23 shows that the highest misclassification occurred between support and parallel. The model classified 64 actual support relationships as parallel and 54 parallels as supports. Regarding the classification of attack labels, the model predicted 29 attack relationships as support and 11 attacks as parallels.

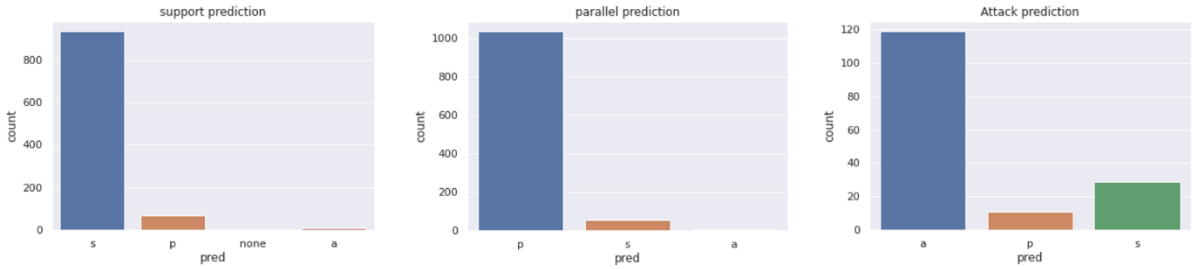


Figure 36 Prediction Results for Each Label

Through manual investigation of these errors, we observed three notable reasons that attribute to the misclassification of the model. The examples of each error are shown in Table 24.

Firstly, we found that misclassification is affected by the loss of context information. This is due to the annotation process where the components in parallel relationships are grouped to either support or attack another component, thus creating contextual information. However, when giving input texts to our NLI model to predict their relationships, only a pair of phrases is given, therefore the context is not fully reflected. Along with this problem, the model wrongly classified the labels due to the emergence of frequently used linguistic patterns in other classes. For instance, phrases in support relations usually contain words that show causal relationships (e.g., ~인 바, ~이므로, ~인 바). To solve these misclassification issues, a possible approach is to provide additional contextual knowledge. For phrases in attack relationships, it can be inferred that the model failed to capture their argumentative relations due to the lack of data. The total number of related argument pairs in our dataset is 12,810. The attack ratio is 0.6 indicating an unbalanced dataset. Due to this lack of attack relations, it can be inferred that the model was not trained enough to correctly distinguish attack phrase pairs,

Table 24 Example Analysis of Support and Parallel Misclassification

Phrase 1	Phrase 2	Golden	Predicted
피고인은 '자신이 119에 전화하여 피해자가 자해를 하였다고 말하였다'고 진술하였다.	위 자백은 그 진술내용이 상당히 구체적이고 자연스러우며 앞서 본 부검소견에도 부합하여 그 신빙성이 높다 할 것이다.	Support	Parallel
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**,	피고인에게는 특수강도의 고의를 인정할 수 **있는바,**	Support	Parallel
그러나 피해자가 이 사건 현장에서 쓰러진 것만으로 앞서 본 바와 같이 전신에 중한 상처를 입었을 것으로는 보이지 않고	피해자가 식칼에 찔린 깊이가 깊고 피해자에게 발생한 자창의 길이도 길다는 점 등을 종합하여 보면,	Parallel	Support
위 상처는 근육이 손상되고 3년간의 후유장애가 있을 정도로 깊은 **상처인바,**	**이로 인하여** 피고인과 F 모두 범인으로 볼 가능성이 존재한다.	Parallel	Support
당시 피고인에게 강취의 고의가 있었음을 넉넉히 인정할 수 있다.	한편 피고인은, 자신의 지시가 없었는데도 피해자가 자진하여 바지에 들어 있던 지갑과 휴대폰을 꺼냈다는 취지로 주장하고 있으나,	Attack	Parallel
피고인으로서는 자신이 수사기관에서 하는 자백의 법적 의미나 그 중요성을 충분히 인식하고 있었을 것인바,	**이에 대하여** 피고인과 변호인은 피고인이 제2, 3회 경찰 피의자신문 및 제1, 2회 검찰 피의자신문 당시 자포자기 심정에서 허위로 자백하였다고 주장하나,	Attack	Support

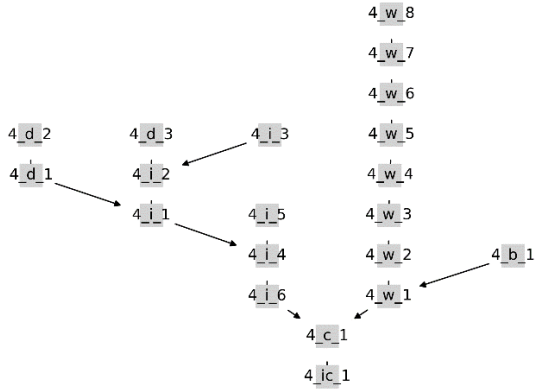
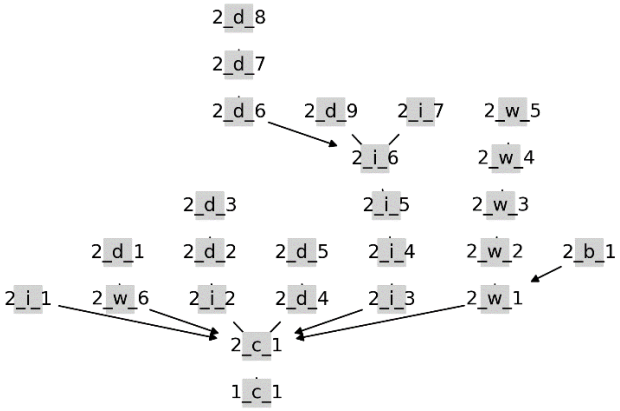
3. Argument structure extraction

The argument structure extraction module generates an argument graph by processing the argument components and their relations. Using this information, the module creates connections based on the respective mappings as (m (in-node), n (out-node)). For this section of the study, we used 221 court decisions as our dataset and divided them into individual argument groups. Therefore, a total of 512 argument units were extracted which were then visualized into tree graphs using our structure extraction module. We use our module to identify the structures and evaluate them. The results show that the extracted graphs are consistent with the formalized visualization patterns.

A. Case Study

To analyze the performance of our system more specifically, we conducted a case study using sample data. The model takes components and their respective related components as input and produces a graph containing this information. The sample predictions from our model are shown in Table 25. From the analysis, we observed that our model can extract the argument structures from the court decisions that fit the graph patterns we have established in Section Ⅲ.

Table 25 An Example of Argument Structure Extraction from our Model

Sentence	Extracted Graph
Case name: 수원지법(여주)2012고합46	
D: 피고인은 …은 사실이나, I: 겁을 주기 위한 것이었을 뿐 C: 살인의 고의는 없었다고 주장한다.	
Case name: 수원지법_2014고합317	
W: 현주건조물방화치사죄가 … 고의가 있어야 하고, I: 이는 … 할 것이며, I: 이 경우 … 할것인데, D: 앞서 본 바와 같이 … 불을 붙였는바, I: 당시 …하였거나, C: 적어도 … 타당하므로, IC: 피고인 및 변호인의 …받아들일 수 없다.	
Case name: 인천지방법원2018고합719	
C: 피해자…고의는 없었다. W: 살인의 … 아니고 W: 자기의 … 충분하다. W: 이때 … 있다. W: 피고인이 … 없었고 W: 단지 … 없다 B: (대법원 …참조). C: 피고인은 …타당하다. I: 피고인이 …이다. D: 피고인은 … 찔렀다. W: 피해자가 … 있다. D: 이 사건 …하였고, D: 많은 양의… 쓰러졌다. I: 피고인은 …판단된다. D: 피고인은 …음에도 D: 피해자를 … 않았다. I: 피고인은 …주었으므로 I: 결과적으로 … 않았고, I: 다른 …주장한다. D: 그러나 … 보았다. D: 이어서 … 가격하였고, D: 이후 …라고 진술한 점, D: 피고인이 … 비추어 보면, I: 피해자 …하였을 뿐 I: 상해를 … 보기 어렵다.	

4. Limitation

While different approaches exist to build a good argument mining model, our experiments have shown that constructing a task-specific legal argument mining corpus is essential to improve the models' performance. For our experiment, court decisions of first-instance criminal courts were used instead of the police investigation reports as access to them was not available to us. The first-instance court decisions share similarities with the police investigation reports as it demonstrates the judge's evaluation of the arguments asserted by the two parties; the defendants and the prosecutor [2]. Based on these similarities, we assume that our approach to using the court decisions as the corpus for extracting legal argument structures suits the purpose of our study which aims to aid the criminal investigation process.

Along with the limitations in our data, despite the promising results of this experiment conducted using our task-specific corpus, we observed that the class imbalance within the dataset hindered the model from accurately predicting the labels. While using the pre-trained language model can improve classification performance, the study can be improved by providing a balanced dataset and applying additional processing to the data that can reduce noise and enhance the model's capability at capturing the context.

Another notable limitation of our study is that we omitted the first task of the general argument mining process which is phrase segmentation. A possible solution to this problem can be approached using a Conditional Random Field (CRF). CRF is a machine learning algorithm that models the dependency between each state and the entire input sequences. Using this characteristic of CRF, texts can be segmented based on the neighboring labels.

VI. Conclusion

To support police investigators in decision-making or in evaluating criminal cases, we propose an argument structure extraction system for the crime investigation process. To this aim, we introduce a novel corpus of first-court decision texts, which are annotated with argumentative components and relations following the argument scheme developed based on the Toulmin argumentation model. Previous approaches regarding this problem are unable to satisfactorily tackle the subtasks of argument mining systems and relied on hand-crafted features which are often time-consuming. We expect that our work will have a significant impact on police investigators as it is a crucial step towards the application of AI to crime investigation.

For this purpose, we employed a Korean pre-trained BERT model to classify argument components. For the relation classification task, we defined it as a multiple-choice problem and further detected their relational stance by using a BERT-based NLI model. To the best of our knowledge, this is the first attempt using transformer-based pre-trained language models for this task in the Korean language. Consequently, the argument structure of the text is extracted and visualized as tree graphs to analyze patterns and evaluate their logicality. In our extensive evaluation, we confirmed that using the Transformer architecture can achieve better performance in every subtask compared to the previous attempts which employed classical machine learning classifiers. We also confirmed that the extracted argument structures that contain information on argument components and relationships correspond to the manually identified argumentative patterns, thus proving that our proposed system can successfully retrieve the internal structures of the legal court decisions.

Finally, we analyzed the errors made by our models. We observed that the misclassification is mainly caused by the loss of contextual information. In future works, we believe that modifying our models to incorporate the knowledge of pronouns or conjunctions used in the text can remedy this problem. Furthermore, we believe that graph-based embeddings can be applied to our model to create a system that can retrieve similar or opposite cases based on the embedding values of argument graphs.

Bibliography

[1] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, and M. Sun, “How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence.” arXiv, May 18, 2020. Accessed: Nov. 21, 2022. [Online]. Available: http://arxiv.org/abs/2004.12158

[2] S. Park, “Alternative hypothesis retrieval model for crime investigation analysis using argument mining,” 국내석사학위논문, 한림대학교, 2020.

[3] “Automatic Extraction and Structure of Arguments in Legal Documents.”

[4] W. Wright, D. Schroh, P. Proulx, A. Skaburskis, and B. Cort, “The Sandbox for analysis: concepts and methods,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Montréal Québec Canada, Apr. 2006, pp. 801–810. doi: 10.1145/1124772.1124890.

[5] Y. B. Shrinivasan and J. J. van Wijk, “Supporting the analytical reasoning process in information visualization,” in *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, Florence, Italy, 2008, p. 1237. doi: 10.1145/1357054.1357247.

[6] B. Verheij, “Automated argument assistance for lawyers,” in *Proceedings of the seventh international conference on Artificial intelligence and law - ICAIL '99*, Oslo, Norway, 1999, pp. 43–52. doi: 10.1145/323706.323714.

[7] F. J. Bex, *Arguments, stories and criminal evidence: a formal hybrid theory*. Dordrecht ; New York: Springer, 2011.

[8] S. W. van den Braak, Cognition and Communication, Intelligent Systems, and Sub Intelligent Systems begr 01-01-2013, *Sensemaking software for crime analysis*. Utrecht University, 2010.

[9] P. Sbarski, T. van Gelder, K. Marriott, D. Prager, and A. Bulka, “Visualizing Argument Structure,” in *Advances in Visual Computing*, vol. 5358, G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, F. Porikli, J. Peters, J. Klosowski, L. Arns, Y. K. Chun, T.-M. Rhyne, and L. Monroe, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 129–138. doi: 10.1007/978-3-

540-89639-5_13.

[10] C. Twardy, “Argument Maps Improve Critical Thinking,” *Teach. Philos.*, vol. 27, no. 2, pp. 95–116, 2004, doi: 10.5840/teachphil200427213.

[11] A. Vaswani *et al.*, “Attention Is All You Need.” arXiv, Dec. 05, 2017. Accessed: Nov. 08, 2022. [Online]. Available: http://arxiv.org/abs/1706.03762

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” arXiv, May 24, 2019. Accessed: Nov. 10, 2022. [Online]. Available: http://arxiv.org/abs/1810.04805

[13] R. M. Palau and M.-F. Moens, “Argumentation mining: the detection, classification and structure of arguments in text,” in *Proceedings of the 12th International Conference on Artificial Intelligence and Law - ICAIL '09*, Barcelona, Spain, 2009, p. 98. doi: 10.1145/1568234.1568246.

[14] M. Lippi and P. Torroni, “Argumentation Mining: State of the Art and Emerging Trends,” *ACM Trans. Internet Technol.*, vol. 16, no. 2, pp. 1–25, Apr. 2016, doi: 10.1145/2850417.

[15] C. Stab and I. Gurevych, “Annotating Argument Components and Relations in Persuasive Essays,” *COLING 2014*, no. The 25th International Conference on Computational Linguistics: Technical Papers, p. 10, 2014.

[16] J. Lawrence and C. Reed, “Argument Mining: A Survey,” *Comput. Linguist.*, vol. 45, no. 4, p. 54, 2019.

[17] J. Lawrence and C. Reed, “Argument Mining Using Argumentation Scheme Structures,” p. 13, 2016.

[18] J. Lawrence and C. Reed, “Combining Argument Mining Techniques,” in *Proceedings of the 2nd Workshop on Argumentation Mining*, Denver, CO, 2015, pp. 127–136. doi: 10.3115/v1/W15-0516.

[19] R. Smith, “Aristotle’s Logic,” in *The Stanford Encyclopedia of Philosophy*, Fall 2020., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2020. Accessed: Sep. 18, 2022. [Online]. Available: https://plato.stanford.edu/archives/fall2020/entries/aristotle-logic/

[20] V. A. Ketcham, “The theory and practice of argumentation and debate,” p. 392, 1914.

[21] J. Fox, P. Krause, and M. EIvang-Gøransson, “Argumentation as a General Framework for Uncertain Reasoning,” in *Uncertainty in Artificial Intelligence*, Elsevier, 1993, pp. 428–434. doi: 10.1016/B978-1-4832-1451-1.50056-1.

[22] I. Habernal and I. Gurevych, “Argumentation Mining in User-Generated Web Discourse,” *Comput. Linguist.*, vol. 43, no. 1, pp. 125–179, Apr. 2017, doi: 10.1162/COLI_a_00276.

[23] C. Stab and I. Gurevych, “Identifying Argumentative Discourse Structures in Persuasive Essays,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 46–56. doi: 10.3115/v1/D14-1006.

[24] J. B. Freeman, *Dialectics and the Macrostructure of Arguments: A Theory of Argument Structure*. DE GRUYTER, 1991. doi: 10.1515/9783110875843.

[25] 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, Mar. 2007, doi: 10.1017/S0269888907001051.

[26] R. Whately, *Elements of logic*, Ninth (Octavo) edition. Whitefish, MT: Kessinger Publishing.

[27] C. Reed and G. Rowe, “A pluralist approach to argument diagramming,” *Law Probab. Risk*, vol. 6, no. 1–4, pp. 59–85, Oct. 2007, doi: 10.1093/lpr/mgm030.

[28] M. C. Beardsley, *Practical Logic*. Prentice-Hall, 1950. [Online]. Available: https://books.google.co.kr/books?id=JJ6uAAAAIAAJ

[29] J. H. Wigmore, *The Principles of Judicial Proof: As Given by Logic, Psychology, and General Experience, and Illustrated in Judicial Trials*. F.B. Rothman, 2000. [Online]. Available: https://books.google.co.kr/books?id=oD9HAQAAMAAJ

[30] S. Toulmin, *The uses of argument*, Updated ed. Cambridge, U.K.; New York: Cambridge University Press, 2003.

[31] A.-H. Tan, “Text Mining: The state of the art and the challenges,” p. 7.

[32] M.-F. Moens, “Argumentation mining: How can a machine acquire common sense and world knowledge?,” *Argum. Comput.*, vol. 9, no. 1, pp. 1–14, Jan. 2018, doi: 10.3233/AAC-170025.

[33] U. Naseem, I. Razzak, S. K. Khan, and M. Prasad, “A Comprehensive Survey on Word Representation Models: From Classical to State-Of-The-Art Word Representation Language Models.” arXiv, Oct. 28, 2020. Accessed: Nov. 01, 2022. [Online]. Available: http://arxiv.org/abs/2010.15036

[34] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. E. Barnes, and D. E. Brown, “Text Classification Algorithms: A Survey,” *Information*, vol. 10, no. 4, p. 150, Apr. 2019, doi: 10.3390/info10040150.

[35] C. C. Aggarwal and C. Zhai, “A Survey of Text Classification Algorithms,” in *Mining Text Data*, C. C. Aggarwal and C. Zhai, Eds. Boston, MA: Springer US, 2012, pp. 163–222. doi: 10.1007/978-1-4614-3223-4_6.

[36] J. Camacho-Collados and M. T. Pilehvar, “On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis.” arXiv, Aug. 23, 2018. Accessed: Oct. 31, 2022. [Online]. Available: http://arxiv.org/abs/1707.01780

[37] Poudyal, “Automatic Extraction and Structure of Arguments in Legal Documents,” University of Évora, 2018.

[38] H. N. Rohman and I. Asror, “Automatic Detection of Argument Components in Text Using Multinomial Nave Bayes Clasiffier,” *J. Phys. Conf. Ser.*, vol. 1192, p. 012034, Mar. 2019, doi: 10.1088/1742-6596/1192/1/012034.

[39] K. S. Jones, *Document Retrieval Systems*, vol. Chapter A Statistical Interpretation of Term Specificity and Its Application in Retrieval. London, UK: Taylor Graham Publishing, 1988.

[40] A. Mandelbaum and A. Shalev, “Word Embeddings and Their Use In Sentence Classification Tasks.” arXiv, Oct. 26, 2016. Accessed: Nov. 03, 2022. [Online]. Available: http://arxiv.org/abs/1610.08229

[41] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality.” arXiv, Oct. 16, 2013. Accessed: Nov. 03, 2022. [Online]. Available: http://arxiv.org/abs/1310.4546

[42] T. Mikolov, W. Yih, and G. Zweig, “Linguistic Regularities in Continuous Space Word Representations,” p. 6.

[43] M. E. Peters *et al.*, “Deep contextualized word representations.” arXiv,

Mar. 22, 2018. Accessed: Nov. 07, 2022. [Online]. Available: http://arxiv.org/abs/1802.05365

[44] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[45] S. Pradhan, K. Hacioglu, V. Krugler, W. Ward, J. H. Martin, and D. Jurafsky, "Support Vector Learning for Semantic Argument Classification," *Mach. Learn.*, vol. 60, no. 1–3, pp. 11–39, Sep. 2005, doi: 10.1007/s10994-005-0912-2.

[46] C. Manning, P. Raghavan, and H. Schuetze, "Introduction to Information Retrieval," p. 581, 2009.

[47] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers," p. 29.

[48] S. Raschka, "Naive Bayes and Text Classification I - Introduction and Theory." arXiv, Feb. 14, 2017. Accessed: Oct. 29, 2022. [Online]. Available: http://arxiv.org/abs/1410.5329

[49] W. Zhang and F. Gao, "An Improvement to Naive Bayes for Text Classification," *Procedia Eng.*, vol. 15, pp. 2160–2164, 2011, doi: 10.1016/j.proeng.2011.08.404.

[50] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," p. 8.

[51] M.-F. Moens, E. Boiy, R. M. Palau, and C. Reed, "Automatic detection of arguments in legal texts," in *Proceedings of the 11th international conference on Artificial intelligence and law - ICAIL '07*, Stanford, California, 2007, p. 225. doi: 10.1145/1276318.1276362.

[52] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.

[53] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks." arXiv, Dec. 14, 2014. Accessed: Nov. 10, 2022. [Online]. Available: http://arxiv.org/abs/1409.3215

[54] A. Saxena, A. Kochsiek, and R. Gemulla, "Sequence-to-Sequence Knowledge Graph Completion and Question Answering." arXiv, Mar. 19, 2022. Accessed: Nov. 10, 2022. [Online]. Available: http://arxiv.org/abs/2203.10321

[55] G. D. Nemeth, “ARGUING WITH BERT: ARGUMENTATION MINING USING CONTEXTUALISED EMBEDDING AND TRANSFORMERS,” p. 101.

[56] Z. Hu, “Question Answering on SQuAD with BERT,” p. 9.

[57] S. Lim, M. Kim, J. Lee, and L. Cns, “KorQuAD1.0 Korean QA Dataset for Machine Reading Comprehension,” p. 5.

[58] A. A. Emelyanov and E. Artemova, “Multilingual Named Entity Recognition Using Pretrained Embeddings, Attention Mechanism and NCRF.” arXiv, Jun. 21, 2019. Accessed: Nov. 10, 2022. [Online]. Available: http://arxiv.org/abs/1906.09978

[59] J. Ham, Y. J. Choe, K. Park, I. Choi, and H. Soh, “KorNLI and KorSTS: New Benchmark Datasets for Korean Natural Language Understanding.” arXiv, Oct. 05, 2020. Accessed: Dec. 12, 2022. [Online]. Available: http://arxiv.org/abs/2004.03289

[60] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding.” arXiv, Feb. 22, 2019. Accessed: Dec. 12, 2022. [Online]. Available: http://arxiv.org/abs/1804.07461

[61] S. Park *et al.*, “KLUE: Korean Language Understanding Evaluation.” arXiv, Nov. 02, 2021. Accessed: Nov. 18, 2022. [Online]. Available: http://arxiv.org/abs/2105.09680

[62] K. Yang, “Transformer-based Korean Pretrained Language Models: A Survey on Three Years of Progress.” arXiv, Nov. 25, 2021. Accessed: Nov. 29, 2022. [Online]. Available: http://arxiv.org/abs/2112.03014

[63] M.-F. Moens, “Argumentation Mining: Where are we now, where do we want to be and how do we get there?,” in *Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation*, New Delhi India, Dec. 2013, pp. 1–6. doi: 10.1145/2701336.2701635.

[64] R. Mochales and M.-F. Moens, “Argumentation mining,” *Artif. Intell. Law*, vol. 19, no. 1, pp. 1–22, Mar. 2011, doi: 10.1007/s10506-010-9104-x.

[65] M. Lippi and P. Torroni, “Argument Mining from Speech: Detecting Claims in Political Debates,” *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, Mar. 2016, doi: 10.1609/aaai.v30i1.10384.

[66] C. Stab and I. Gurevych, “Parsing Argumentation Structures in Persuasive Essays,” *Comput. Linguist.*, vol. 43, no. 3, pp. 619–659, Sep. 2017, doi: 10.1162/COLI_a_00295.

[67] R. Duthie, K. Budzynska, and C. Reed, “Mining Ethos in Political Debate,” p. 12.

[68] R. Levy, Y. Bilu, D. Hershcovich, E. Aharoni, and N. Slonim, “Context Dependent Claim Detection,” p. 12.

[69] O. Biran and O. Rambow, “IDENTIFYING JUSTIFICATIONS IN WRITTEN DIALOGS BY CLASSIFYING TEXT AS ARGUMENTATIVE,” *Int. J. Semantic Comput.*, vol. 05, no. 04, pp. 363–381, Dec. 2011, doi: 10.1142/S1793351X11001328.

[70] E. Cabrio and S. Villata, “Combining Textual Entailment and Argumentation Theory for Supporting Online Debates Interactions,” p. 5.

[71] V. Niculae, J. Park, and C. Cardie, “Argument Mining with Structured SVMs and RNNs,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, 2017, pp. 985–995. doi: 10.18653/v1/P17-1091.

[72] G. Morio, H. Ozaki, T. Morishita, Y. Koreeda, and K. Yanai, “Towards Better Non-Tree Argument Mining: Proposition-Level Biaffine Parsing with Task-Specific Parameterization,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 3259–3266. doi: 10.18653/v1/2020.acl-main.298.

[73] C. Stab and I. Gurevych, “Parsing Argumentation Structures in Persuasive Essays.” arXiv, Jul. 22, 2016. Accessed: Sep. 04, 2022. [Online]. Available: http://arxiv.org/abs/1604.07370

[74] E. Cabrio and S. Villata, “Five Years of Argument Mining: a Data-driven Analysis,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, Jul. 2018, pp. 5427–5433. doi: 10.24963/ijcai.2018/766.

[75] A. Lytos, T. Lagkas, P. Sarigiannidis, and K. Bontcheva, “The evolution of argumentation mining: From models to social media and emerging tools,” *Inf. Process. Manag.*, vol. 56, no. 6, p. 102055, Nov. 2019, doi: 10.1016/j.ipm.2019.102055.

[76] J. Eckle-Kohler, R. Kluge, and I. Gurevych, “On the Role of Discourse Markers for Discriminating Claims and Premises in Argumentative Discourse,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp. 2236–2242. doi: 10.18653/v1/D15-1267.

[77] T. Goudas, C. Louizos, and G. Petasis, “Argument Extraction from News, Blogs, and Social Media,” in *SETN 2014*, Cham, 2014, vol. 8445. doi: 10.1007/978-3-319-07064-3.

[78] E. Aharoni *et al.*, “A Benchmark Dataset for Automatic Detection of Claims and Evidence in the Context of Controversial Topics,” in *Proceedings of the First Workshop on Argumentation Mining*, Baltimore, Maryland, 2014, pp. 64–68. doi: 10.3115/v1/W14-2109.

[79] R. Rinott, L. Dankin, C. Alzate Perez, M. M. Khapra, E. Aharoni, and N. Slonim, “Show Me Your Evidence - an Automatic Method for Context Dependent Evidence Detection,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp. 440–450. doi: 10.18653/v1/D15-1050.

[80] B. Birnbaum, M. Flowers, and R. McGuire, “Towards an AI model of argumentation,” in *AAAI Press*, 1980, pp. 313–315.

[81] I. Habernal, J. Eckle-Kohler, and I. Gurevych, “Argumentation Mining on the Web from Information Seeking Perspective,” p. 14.

[82] N. Nguyen and Y. Guo, “Comparisons of sequence labeling algorithms and extensions,” in *Proceedings of the 24th international conference on Machine learning - ICML '07*, Corvalis, Oregon, 2007, pp. 681–688. doi: 10.1145/1273496.1273582.

[83] L. Getoor, “Tutorial on Statistical Relational Learning,” in *Inductive Logic Programming*, vol. 3625, S. Kramer and B. Pfahringer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 415–415. doi: 10.1007/11536314_26.

[84] C. Sardianos, I. M. Katakis, G. Petasis, and V. Karkaletsis, “Argument Extraction from News,” in *Proceedings of the 2nd Workshop on Argumentation Mining*, Denver, CO, 2015, pp. 56–66. doi: 10.3115/v1/W15-0508.

[85] N. Kwon, E. Hovy, L. Zhou, and S. W. Shulman, “Identifying and Classifying Subjective Claims,” p. 6.

[86] M. Lippi and P. Torroni, “Context-Independent Claim Detection for Argument Mining,” p. 7.

[87] N. Reimers, B. Schiller, T. Beck, J. Daxenberger, C. Stab, and I. Gurevych, “Classification and Clustering of Arguments with Contextualized Word Embeddings.” arXiv, Jun. 24, 2019. Accessed: Sep. 29, 2022. [Online]. Available: http://arxiv.org/abs/1906.09821

[88] T. Chakrabarty, C. Hidey, S. Muresan, K. McKeown, and A. Hwang, “AMPERSAND: Argument Mining for PERSuAsive oNline Discussions,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 2933–2943. doi: 10.18653/v1/D19-1291.

[89] R. Ruiz-Dolz, S. Heras, J. Alemany, and A. García-Fornes, “Transformer-Based Models for Automatic Identification of Argument Relations: A Cross-Domain Evaluation,” *IEEE Intell. Syst.*, vol. 36, no. 6, pp. 62–70, Nov. 2021, doi: 10.1109/MIS.2021.3073993.

[90] S. Eger, J. Daxenberger, and I. Gurevych, “Neural End-to-End Learning for Computational Argumentation Mining,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, 2017, pp. 11–22. doi: 10.18653/v1/P17-1002.

[91] M. P. García-Villalba and P. Saint-Dizier, “A Framework to Extract Arguments in Opinion Texts:,” *Int. J. Cogn. Inform. Nat. Intell.*, vol. 6, no. 3, pp. 62–87, Jul. 2012, doi: 10.4018/jcini.2012070104.

[92] M. Lippi and P. Torroni, “MARGOT: A web server for argumentation mining,” *Expert Syst. Appl.*, vol. 65, pp. 292–303, Dec. 2016, doi: 10.1016/j.eswa.2016.08.050.

[93] N. Stylianou and I. Vlahavas, “TransforMED: End-to-End Transformers for Evidence-Based Medicine and Argument Mining in medical literature,” *J. Biomed. Inform.*, vol. 117, p. 103767, May 2021, doi: 10.1016/j.jbi.2021.103767.

[94] T. Mayer, E. Cabrio, and S. Villata, “Transformer-based Argument Mining for Healthcare Applications,” p. 9.

[95] M. Teruel, C. Cardellino, F. Cardellino, L. A. Alemany, and S. Villata, “Increasing Argument Annotation Reproducibility by Using Inter-annotator

Agreement to Improve Guidelines," p. 4.

[96] K. A. Khatib, H. Wachsmuth, J. Kiesel, M. Hagen, and B. Stein, "A News Editorial Corpus for Mining Argumentation Strategies," p. 11.

[97] D. N. Walton, C. Reed, and F. Macagno, *Argumentation schemes*. Cambridge ; New York: Cambridge University Press, 2008.

[98] V. W. Feng and G. Hirst, "Classifying arguments by scheme," p. 10.

[99] G. Rowe, F. Macagno, C. Reed, D. Walton, and Philosophy Documentation Center, "Araucaria as a Tool for Diagramming Arguments in Teaching and Studying Philosophy:," *Teach. Philos.*, vol. 29, no. 2, pp. 111–124, 2006, doi: 10.5840/teachphil200629217.

[100] J. Lawrence, F. Bex, C. Reed, and M. Snaith, "AIFdb: Infrastructure for the Argument Web," p. 2.

[101] P. Poudyal, J. Savelka, A. Ieven, M. F. Moens, T. Goncalves, and P. Quaresma, "ECHR: Legal Corpus for Argument Mining," p. 9.

[102] R. Mochales and M.-F. Moens, "Study on the Structure of Argumentation in Case Law," p. 11.

[103] 백상준, "Current Status and Future Tasks of the Online Access to Court Records System [판결서 인터넷열람 제도의 개선현황과 향후과제]," 이슈와 논점, no. 1571, p. 4, 2019.

[104] Y. H. Kim, "Application of Text Mining for Legal Information System: Focusing on Defamation Precedent," *J. Korean Soc. Libr. Inf. Sci.*, vol. 54, no. 1, pp. 387–409, 28 2020, doi: 10.4275/KSLIS.2020.54.1.387.

[105] K. Krippendorff, "Measuring the Reliability of Qualitative Text Analysis Data," *Qual. Quant.*, vol. 38, no. 6, pp. 787–800, Dec. 2004, doi: 10.1007/s11135-004-8107-7.

[106] 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, May 2020, doi: 10.1017/dsd.2020.153.

[107] J. L. Fleiss, "Measuring nominal scale agreement among many raters.," *Psychol. Bull.*, vol. 76, no. 5, pp. 378–382, Nov. 1971, doi: 10.1037/h0031619.

[108] M. Miwa and M. Bansal, “End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, 2016, pp. 1105–1116. doi: 10.18653/v1/P16-1105.

[109] J. Li, E. Durmus, and C. Cardie, “Exploring the Role of Argument Structure in Online Debate Persuasion,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, 2020, pp. 8905–8912. doi: 10.18653/v1/2020.emnlp-main.716.

[110] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, “SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference.” arXiv, Aug. 15, 2018. Accessed: Nov. 18, 2022. [Online]. Available: http://arxiv.org/abs/1808.05326

[111] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.” arXiv, Feb. 29, 2020. Accessed: Nov. 26, 2022. [Online]. Available: http://arxiv.org/abs/1910.01108

[112] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, “A large annotated corpus for learning natural language inference.” arXiv, Aug. 21, 2015. Accessed: Nov. 18, 2022. [Online]. Available: http://arxiv.org/abs/1508.05326

[113] I. Dagan, O. Glickman, and B. Magnini, “The PASCAL Recognising Textual Entailment Challenge,” in *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, vol. 3944, J. Quiñonero-Candela, I. Dagan, B. Magnini, and F. d’Alché-Buc, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 177–190. doi: 10.1007/11736790_9.

[114] A. Williams, N. Nangia, and S. R. Bowman, “A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference.” arXiv, Feb. 19, 2018. Accessed: Nov. 18, 2022. [Online]. Available: http://arxiv.org/abs/1704.05426

[115] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, “Supervised Learning of Universal Sentence Representations from Natural Language Inference Data.” arXiv, Jul. 08, 2018. Accessed: Nov. 18, 2022. [Online]. Available: http://arxiv.org/abs/1705.02364

[116] A. Poliak, Y. Belinkov, J. Glass, and B. Van Durme, “On the Evaluation of Semantic Phenomena in Neural Machine Translation Using Natural Language Inference,” in *Proceedings of the 2018 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, Louisiana, 2018, pp. 513–523. doi: 10.18653/v1/N18-2082.

[117] T. F. Gordon, "Analyzing open source license compatibility issues with Carneades," in *Proceedings of the 13th International Conference on Artificial Intelligence and Law - ICAIL '11*, Pittsburgh, Pennsylvania, 2011, pp. 51–55. doi: 10.1145/2018358.2018364.

[118] D. N. Walton and T. F. Gordon, "The Carneades model of argument invention," *Pragmat. Cogn.*, vol. 20, no. 1, pp. 1–31, May 2012, doi: 10.1075/pc.20.1.01wal.

[119] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring Network Structure, Dynamics, and Function using NetworkX," p. 5, 2008.

[120] L. Tu, G. Lalwani, S. Gella, and H. He, "An Empirical Study on Robustness to Spurious Correlations using Pre-trained Language Models," *Trans. Assoc. Comput. Linguist.*, vol. 8, pp. 621–633, Dec. 2020, doi: 10.1162/tacl_a_00335.

[121] S. Sagawa, A. Raghunathan, P. W. Koh, and P. Liang, "An Investigation of Why Overparameterization Exacerbates Spurious Correlations." arXiv, Aug. 26, 2020. Accessed: Nov. 20, 2022. [Online]. Available: http://arxiv.org/abs/2005.04345

범죄 수사를 위한 트랜스포머(Transformer) 모델 기반 법적 논증 구조 추출 모델에 관한 연구

2022.

석사학위논문

구예리

국제학과

지도교수: 박노섭, 노기영

2021년 개정된 형사소송법 및 검찰청법의 시행으로 인해 수사환경이 변화함에 따라 경찰에게 1차적인 수사를 책임져야 할 주체적 지위를 부여하여, 경찰 수사관 측 사건 검토 과정이 유례없이 중요해진 상황을 맞이했다. 또한 수사종결권 부여와 함께 공개된 법정에서 직접 증거 조사를 강화하는 공판중심주의가 강화되는 추세에 따라 객관적 증거를 기반으로 한 법정에서의 논리적 사실주장이 더욱 요청되는 바이다. 이에 논증을 통한 수사검증은 경찰에 요구되는 핵심역량이 될 것임을 예측할 수 있다. 그러나 기존의 사건 분석 지원 도구는 논리적인 검증 보다는 증거 수집 및 분석에 중점을 두고 있는 현황이며, 논리적 완결성을 갖춘 사건분석 및 검토를 위해서는 증거를 기반으로 법적 논리적 쟁점을 도출할 수 있는 논증 분석 시스템이 요구된다. 본 연구의 목적은 논증마이닝 기법을 통해 입력된 수사문서에서 (1) 논증을 구성하는 요소를 자동으로 추출하고, (2) 추출된 논증요소들 간의 관계를 자동으로 분류하여 (3) 자동 추출된 논증구조 그래프를 제공함으로써 수사 관계자들이 빠르고 객관적인 시각으로 사건의 논증 구조를 검토할 수 있는 모델 구조를 고안하는 데 있다. 또한 이러한 일련의 과정에 최근 자연어 처리 분야에서 활발히 사용되고 있는 트랜스포머 (Tranformer) 기반 사전학습 언어모델을 사용하여 그 성능을 높이고자 한다.

논증 마이닝 (Argumentation Mining)은 자연어 처리 연구 분야의 일부로서 텍스트에서 논증을 식별하고 분석하는 것을 목적으로 하며 교육, 정책, 소셜 미디어 (social media) 및 법률을 비롯한 다양한 도메인에서 사용되고 있다. 본 연구에서는 현행 수사결과보고서와 유사한 구조를 가지는 제1심 형사판결문 256건을 대상으로 툴민(Toulmin)의 논증구조를 본 연구의 목적에 맞게 확장 및 재개념한 논증 모델인 Toulmin+ 모델을 기반으로 논증 요소와 관계를 분석하여 논증 마이닝의 데이터로 활용하였다.

본 연구의 첫번째 과제인 논증 요소 자동 추출은 자연어 처리 연구 분야의

다양한 태스크에서 우수한 성능을 보이고 있는 트랜스포머 (Transformer) 기반 사전학습 모델인 BERT의 한국어 모델 KoBERT를 사용하여 총 7가지의 논증 요소에 대한 다중 분류를 시도하였다. 실험 결과 관련 선행연구에서 좋은 성적을 보인 지도학습 기반 문장 분류 기법인 Support Vector Machine과 대등한 성능을 보여줌으로써 사전학습 모델을 법률 데이터에 파인튜닝 (Fine-tuning) 할 수 있음을 확인하였다. 두번째 과제인 논증 관계 추출은 한국어 트랜스포머 기반 사전학습 모델인 Klue-BERT base의 BertForMultipleChoice 모델과 자연어 추론 모델인 NLI 모델을 활용하여 문서 내 관련 구절 후보군 중 가장 높은 연관도를 지닌 구절을 추출하고, 두 문장간의 관계성을 분류하였다. 실험 결과 우수한 성능을 보여줌으로서 다양한 선행연구에서 지적한 논증 관계 추출의 어려움을 사전학습 모델을 사용함으로서 해결할 수 있음을 보여주었다는 점에서 큰 의의가 있다. 마지막으로 본 연구는 두 가지의 선행 과제를 통해 새 문서의 논증 요소와 관계를 식별하고, 이를 바탕으로 논증 구조를 생성하여 이를 그래프 형태로 시각화하는 것을 목표로 한다. 본 모델을 적용하여 시각화 된 논증 구조를 도출한 결과, 판결문에 등장하는 논증 구조들의 특정 유형이 존재하고 이를 본 연구에서 사용하는 모델을 기반으로 나타낼 수 있음을 확인할 수 있었다.

본 연구는 추가적인 기술 개선을 통해 논증 구조 추출 뿐만 아니라 추출된 논증 그래프를 임베딩하여 학습한 모델을 활용해 유사 사건 검색 및 비교분석 등 인공지능 수사 시스템의 다양한 분야에 사용될 수 있을 것으로 기대된다.

주제어: 범죄수사, 논증 마이닝, 트랜스포머, 논증 구조 자동 추출 모델, 논증 시각화

Transformer-based Legal Argument Structure Extraction Model

2022.

Master's Degree

Gu, Yeri

Department of International Studies

Advisor Prof. Park, Roh Sub, Prof. Noh, Ghee Young

The implementation of the revised Korean Criminal Procedure Act in 2020 grants a subjective position to the police to be responsible for the primary investigation, thus making the police investigator's case review process unprecedentedly important. In addition, newly amended legislation strengthens the direct investigation of evidence in courts, hence, logical proving cases in court based on objective evidence are further requested. With such change, the verification of the investigation process through argumentation is expected to be a core competency required by the police. However, the existing case analysis tools focus on collecting and analyzing evidence rather than logical verification, therefore, an argument analysis system that can derive legal claims based on evidence is required for case analysis with logical completeness. The purpose of this study is to devise an argument mining model that allows investigators to examine the case's argument structure with a quick and objective perspective by (1) automatically extracting the argument components, and (2) classifying the relationship between the extracted argument pairs. We also aim to increase the model's performance by using Transformer-based architectures, which have recently been actively used in the field of natural language processing. Argument Mining is an NLP method that identifies arguments in text and is used in various domains, including education, policy, social media, and law. In this study, 256 criminal judgments of the first court were used to analyze argument components and relations based on the Toulmin+ argument model which is an expanded and reconceived version of the original Toulmin model.

The first task of this study attempts to multi-classify a total of seven argument components using the Korean BERT model. The results confirmed that the pre-

trained model can be fine-tuned to the legal corpus by showing equivalent performance to the Support Vector Machine, a supervised classification method that performed well in previous studies. The second task uses the BertForMultipleChoice model and the KLUE BERT-base NLI model to extract the most related phrase in the document and classify their relationships. The model's outstanding performance is significant considering the difficulty of extracting argument relationships pointed out in previous studies. Finally, this study proposes a system that extracts the argument structures through two preceding tasks and visualizes them in graph form. The results showed that a specific type of argument structure exists in court decisions and that they can be expressed through the model developed in this study.

This study is expected to be used in various fields of artificial intelligence investigation systems such as similar case retrieval by training the model on the extracted argument graphs embeddings through additional technological improvements.

Keywords: Crime investigation, Argument mining, Transformer, Automatic argument structure extraction model, Argument Visualization

Appendix

<Appendix 1> Example Court Decision Annotation

창원지법마산지원 2019고합40살인


```
창원지법마산지원 2019고합40살인
{
    "meta":
    {
        "case_id": 83,
        "title": "창원지방법원마산지원2019고합40",
        "type": "murder"
    },
    "annotation_data" : [
        {
            "toulmin_No": 1,
            "component": "1_i_1",
            "relation": "1_c_1",
            "relation_type": "a",
            "defeated": "Y",
            "phrase": "피고인은 조현병으로 환청, 망상 등에 사로잡혀 사물을 변별할 능력이 없거나 의사를 결정할 능력이 없는 상태에서 이 사건 범행을 저질렀으므로,"
        },
        {
            "toulmin_No": 1,
            "component": "1_c_1",
            "relation": "2_c_1",
            "relation_type": "a",
            "defeated": "Y",
            "phrase": "피고인에게 형사책임을 물을 수 없다."
        },
        {
            "toulmin_No": 2,
            "component": "2_d_1",
```

```
            "relation": "2_i_1",
            "relation_type": "s",
            "defeated": "N",
            "phrase": "피고인이 이 사건 범행 당시 조현병으로 사물을 변별할 능력이나 의사를 결정할 능력이 미약한 상태에서 있었음은 앞서 본 바와 같다."
        },
        {
            "toulmin_No": 2,
            "component": "2_d_2",
            "relation": "2_d_1",
            "relation_type": "p",
            "defeated": "N",
            "phrase": "그러나 이 법원이 적법하게 채택하여 조사한 증거들에 의하여 인정되는 사정, 즉"
        },
        {
            "toulmin_No": 2,
            "component": "2_i_1",
            "relation": "2_c_1",
            "relation_type": "s",
            "defeated": "N",
            "phrase": "피고인이 수사기관에서 이 사건 범행의 내용을 상당히 구체적으로 진술한 점,"
        },
        {
            "toulmin_No": 2,
            "component": "2_i_2",
            "relation": "2_i_1",
            "relation_type": "p",
            "defeated": "N",
            "phrase": "피고인이 이 사건 당시 범행이 미칠 영향에 대하여도 인식하고 있었던 것으로 보이는 점,"
        },
        {
            "toulmin_No": 2,
            "component": "2_d_3",
            "relation": "2_d_2",
            "relation_type": "p",
            "defeated": "N",
            "phrase": "그밖에 이 사건 범행의 경위, 수단과 방법, 범행 후 피고
```

```
인의 행동 등 제반 사정을 종합해 보면,"
        },
        {
            "toulmin_No": 2,
            "component": "2_c_1",
            "relation": "2_ic_1",
            "relation_type": "s",
            "defeated": "N",
            "phrase": "피고인은 이 사건 범행 당시 조현병으로 인하여 사물을 변별하거나 의사를 결정할 능력이 미약한 상태에서 더 나아가 심신상실 상태에 있었다고 보기 어렵다."
        },
        {
            "toulmin_No": 2,
            "component": "2_ic_1",
            "relation": "",
            "relation_type": "",
            "defeated": "N",
            "phrase": "따라서 위 주장은 받아들이기 어렵다."
        }
    ]
}
```

<Appendix 2> List of Annotated Corpus

Number	Case ID	Case Name
0	B101	광주지법(목포)2015 고합 19
1	B102	광주지법(목포)2018 고합 36
2	B105	광주지법(순천)2020 고합 224
3	B107	광주지법 2013 고합 544
4	B108	광주지법 2013 고합 85
5	B109	광주지법 2017 고합 307
6	B110	광주지법 2018 고합 519
7	B111	광주지법 2019 고합 446
8	B112	대구지법(경주)2019 고합 8
9	B113	대구지법(김천)2012 고합 114
10	B114	대구지법(김천)2014 고합 87
11	B116	대구지법(김천)2018 고합 26
12	B117	대구지법(상주)2014 고합 39
13	B118	대구지법(서부)2013 고합 140
14	B119	대구지법(서부)2013 고합 52
15	B11	부산지방법원동부지원 2018 고합 110 판결
16	B120	대구지법(안동)2020 고합 19
17	B121	대구지법(영덕)2014 고합 1
18	B122	대구지법(영덕)2017 고합 3
19	B124	대구지법(포항)2016 고합 78
20	B125	대구지법 2012 고합 450
21	B126	대구지법 2015 고합 15
22	B127	대구지법 2015 고합 195
23	B128	대구지법 2017 고합 414
24	B129	대구지법 2017 고합 514
25	B130	대구지법 2018 고합 234
26	B131	대구지법 2018 고합 51
27	B133	대구지법 2020 고합 7

28	B134	대전지법(논산)2014 고합 2
29	B135	대전지법(논산)2019 고합 55
30	B136	대전지법(천안)2016 고합 86
31	B137	대전지법(천안)2017 고합 39
32	B138	대전지법(홍성)2013 고합 42
33	B139	대전지법(홍성)2014 고합 56
34	B140	대전지법(홍성)2014 고합 70
35	B141	대전지법(홍성)2015 고합 15
36	B142	대전지법(홍성)2019 고합 25
37	B143	대전지법 2012 고합 382
38	B144	대전지법 2012 고합 400
39	B145	대전지법 2013 고합 139
40	B146	대전지법 2013 고합 513
41	B148	대전지법 2015 고합 142
42	B149	대전지법 2016 고합 347
43	B14	부산지방법원 2008 고합 143
44	B150	대전지법 2017 고합 208
45	B151	대전지법 2018 고합 150
46	B153	대전지법 2019 고합 207
47	B154	대전지법 2019 고합 232
48	B155	대전지법 2020 고합 167
49	B156	대전지법 2020 고합 61
50	B157	대전지법 2020 고합 92
51	B158	부산지법(동부)2018 고합 155
52	B159	부산지법(서부)2018 고합 112
53	B15	대전지방법원 2004 고합 367 판결
54	B160	부산지법 2012 고합 423
55	B161	부산지법 2012 고합 538
56	B162	부산지법 2013 고합 146
57	B163	부산지법 2013 고합 338
58	B164	부산지법 2016 고합 586
59	B165	부산지법 2016 고합 828
60	B167	서울남부지법 2013 고합 66
61	B168	서울남부지법 2014 고합 274

62	B169	서울남부지법 2015 고합 354
63	B16	대전지방법원 2012 고합 31 판결
64	B171	서울남부지법 2018 고합 594
65	B172	서울남부지법 2019 고합 191
66	B175	서울동부지법 2014 고합 243
67	B176	서울동부지법 2016 고합 132
68	B177	서울동부지법 2017 고합 155
69	B178	서울동부지법 2018 고합 366
70	B179	서울북부지법 2012 고합 371
71	B17	대전지방법원 2012 고합 380 판결
72	B180	서울북부지법 2014 고합 364
73	B181	서울북부지법 2015 고합 227
74	B182	서울북부지법 2017 고합 316
75	B183	서울북부지법 2017 고합 490
76	B184	서울북부지법 2018 고합 393
77	B185	서울북부지법 2020 고합 143
78	B186	서울북부지법 2020 고합 15
79	B187	서울서부지법 2017 고합 375
80	B188	서울서부지법 2019 고합 211
81	B190	서울중앙지법 2012 고합 1314
82	B191	서울중앙지법 2013 고합 91
83	B192	서울중앙지법 2015 고합 189
84	B193	서울중앙지법 2015 고합 227
85	B194	서울중앙지법 2015 고합 785
86	B195	서울중앙지법 2016 고합 869
87	B196	서울중앙지법 2018 고합 151
88	B198	서울중앙지법 2018 고합 159
89	B199	서울중앙지법 2018 고합 503
90	B19	대전지방법원 2018 고합 353 판결
91	B1	인천지방법원 2018 고합 17 판결
92	B200	서울중앙지법 2018 고합 839
93	B201	서울중앙지법 2019 고합 862
94	B202	서울중앙지법 2020 고합 262
95	B203	수원지법(성남)2014 고합 219

96	B204	수원지법(성남)2015 고합 71
97	B205	수원지법(성남)2016 고합 301
98	B206	수원지법(성남)2017 고합 156
99	B207	수원지법(성남)2017 고합 160
100	B208	수원지법(성남)2017 고합 288
101	B210	수원지법(안산)2014 고합 79
102	B211	수원지법(안산)2015 고합 130
103	B212	수원지법(안산)2015 고합 160
104	B213	수원지법(안산)2018 고합 289
105	B214	수원지법(안양)2016 고합 221
106	B216	수원지법(안양)2019 고합 47
107	B217	수원지법(여주)2012 고합 46
108	B219	수원지법(평택)2016 재고합 3
109	B21	대전지방법원 2018 고합 452 판결
110	B220	수원지법 2012 고합 1119
111	B221	수원지법 2012 고합 485
112	B222	수원지법 2013 고합 103
113	B223	수원지법 2013 고합 599
114	B224	수원지법 2014 고합 317
115	B225	수원지법 2015 고합 505
116	B226	수원지법 2016 고합 364
117	B228	수원지법 2016 고합 644
118	B229	수원지법 2016 고합 738
119	B22	대전지방법원 2019 고합 110
120	B231	수원지법 2019 고합 497
121	B233	울산지법 2015 고합 229
122	B235	울산지법 2017 고합 168
123	B236	울산지법 2017 고합 218
124	B237	울산지법 2020 고합 131
125	B239	의정부지법고양 2019 고합 204
126	B23	대전지방법원논산지원 2018 고합 36 판결
127	B240	의정부지법 2013 고합 538
128	B241	의정부지법 2014 고합 249
129	B242	의정부지법 2014 고합 359

130	B243	의정부지법 2015 고합 369
131	B244	의정부지법 2020 고합 113
132	B246	인천지법 2012 고합 1058
133	B248	인천지법 2015 고합 518
134	B249	인천지법 2015 고합 599
135	B24	대전지방법원천안지원 2018 고합 240 판결
136	B253	인천지법 2017 고합 548
137	B256	인천지법 2019 고합 429
138	B257	인천지법 2019 고합 630
139	B258	인천지법 2020 고합 387
140	B259	전주지법(군산)2018 고합 48
141	B25	대전지방법원홍성지원 2018 고합 91 판결
142	B260	전주지법(남원)2014 고합 21
143	B261	전주지법(정읍)2012 고합 122
144	B262	전주지법 2012 고합 378
145	B264	전주지법 2014 고합 306
146	B266	전주지법 2020 고합 58
147	B267	제주지법 2012 고합 307
148	B268	제주지법 2014 고합 179
149	B269	제주지법 2019 고합 133
150	B26	대구지방법원 2005 고합 623 판결
151	B270	제주지법 2020 고합 7
152	B271	창원지법(마산)2015 고합 45
153	B272	창원지법(마산)2016 고합 124
154	B273	창원지법(마산)2016 고합 14
155	B274	창원지법(마산)2016 고합 89
156	B275	창원지법(마산)2019 고합 102
157	B276	창원지법(밀양)2017 고합 2
158	B277	창원지법(밀양)2019 고합 13
159	B278	창원지법(진주)2017 고합 10
160	B27	대구지방법원 2006 고합 36
161	B280	창원지법(진주)2017 고합 55
162	B281	창원지법(진주)2018 고합 93
163	B282	창원지법(통영)2013 고합 23

164	B283	창원지법(통영)2014 고합 47
165	B285	창원지법(통영)2016 고합 40
166	B286	창원지법 2016 고합 194
167	B288	창원지법 2020 고합 55
168	B289	청주지법(영동)2017 고합 20
169	B290	청주지법(충주)2016 고합 9
170	B291	청주지법(충주)2018 고합 53
171	B293	청주지법 2018 고합 125
172	B294	청주지법 2019 고합 33
173	B295	청주지법 2020 고합 61
174	B296	춘천지법(속초)2016 고합 45
175	B298	춘천지법(영월)2019 고합 11
176	B299	춘천지법 2017 고합 19
177	B29	대구지방법원 2006 고합 667 판결
178	B2	전주지방법원군산지원 2017 고합 21 판결
179	B300	춘천지법 2020 고합 10
180	B3	창원지방법원 2020 고합 18
181	B401	서울동부지법 2018 고합 55
182	B402	서울동부지법 2020 고합 102
183	B403	서울동부지법 2020 고합 222
184	B404	서울북부지법 2012 고합 631
185	B405	서울북부지법 2015 고합 180
186	B406	서울북부지법 2015 고합 323
187	B407	서울북부지법 2016 고합 151
188	B408	서울북부지법 2016 고합 166
189	B409	서울북부지법 2016 고합 269
190	B410	서울북부지법 2016 고합 346
191	B413	서울북부지법 2018 고합 426
192	B414	서울북부지법 2018 고합 466
193	B416	서울북부지법 2021 고합 92
194	B418	서울서부지법 2017 고합 129
195	B419	서울서부지법 2017 고합 396
196	B420	서울서부지법 2019 고합 340
197	B4	부산지방법원 2010 고합 372 판결

198	B5	서울동부지방법원 2010 고합 348
199	B6	수원지방법원 2006 고합 1 판결
200	B72	울산지법 2014.3.14 선고 2013 고합 311 판결
201	B77	창원지법 2009.9.16 선고 2009 고합 94 판결
202	B84	울산지방법원 2014 고합 235
203	B85	수원지방법원 2006 고합 1 판결
204	B88	인천지방법원 2018 고합 17 판결
205	B8	춘천지방법원 2012 고합 63
206	B95	전주지방법원 2009 고합 62 판결
207	B99	제주지방법원 2010 고합 99 판결
208	B9	부산지방법원 2019 고합 420
209	B301	광주지법(목포)2015 고합 18
210	B302	광주지법(목포)2015 고합 30
211	B303	광주지법(목포)2019 고합 38
212	B304	광주지법(목포)2019 고합 41
213	B305	광주지법(순천)2013 고합 80
214	B306	광주지법(순천)2013 고합 87
215	B307	광주지법(순천)2015 고합 216
216	B308	광주지법(순천)2019 고합 30
217	B309	광주지법(순천)2019 고합 94
218	B310	광주지법(순천)2020 고합 217
219	B311	광주지법(해남)2016 고합 41
220	B312	광주지법 2012 고합 1200
221	B313	광주지법 2013 고합 420
222	B314	광주지법 2014 고합 298
223	B315	광주지법 2017 고합 156
224	B316	광주지법 2017 고합 434
225	B318	광주지법 2019 고합 249
226	B320	광주지법 2019 고합 75
227	B321	대구지법(경주)2014 고합 27
228	B323	대구지법(김천)2014 고합 91
229	B324	대구지법(김천)2019 고합 8
230	B325	대구지법(서부)2012 고합 488
231	B326	대구지법(서부)2013 고합 223

232	B328	대구지법(서부)2020 고합 17
233	B329	대구지법(의성)2015 고합 17
234	B330	대구지법(의성)2016 고합 30
235	B331	대구지법 2013 고합 329
236	B332	대구지법 2013 고합 541
237	B333	대구지법 2014 고합 87
238	B334	대구지법 2015 고합 184
239	B335	대구지법 2015 고합 457 판결서
240	B336	대구지법 2016 고합 114
241	B361	대전지법(홍성)2020 고합 3
242	B362	대전지법(홍성)2020 고합 48
243	B363	대전지법 2013 고합 356
244	B364	대전지법 2013 고합 426
245	B367	대전지법 2016 고합 334
246	B368	대전지법 2017 고합 430
247	B370	대전지법 2018 고합 281
248	B371	대전지법 2018 고합 405
249	B372	대전지법 2019 고합 160
250	B373	대전지법 2019 고합 319
251	B374	대전지법 2019 고합 411
252	B375	대전지법 2019 고합 426
253	B376	대전지법 2020 고합 116
254	B378	부산지법(동부)2013 고합 40
255	B379	부산지법(동부)2013 고합 84
256	B380	부산지법(동부)2018 고합 135

<Appendix 3> Sample Case Annotation

Case name: 수원지법(안양)2018고합110

1) Component Classification

Phrase	Component	Prediction
피고인은 이 법정에서 범행을 모두 인정하면서도 한편으로 반성문을 통하여 '피해자가 문을 열어주어서 들어갔을 뿐, 피해자의 주거에 침입하지는 아니하였다'는 취지로 주장한다	c	i
살피건대, 아래와 같은 피해자 및 관련자들의 진술에 의하면,	u	u
피고인이 피해자 주거지의 현관문을 불상의 방법으로 열거나 번호키를 해제하는 방법으로 피해자의 주거에 침입한 사실을 충분히 인정할 수 있으므로	c	c
피고인의 주장은 받아들이기 어렵다.	ic	ic
1 피해자는 수사기관에서 2018. 2.경 범행에 관하여	d	d
"누가 현관문을 두드리더니,뒷창문으로 (집 안을 들여다) 보면서 문 뜯고 들어가겠다고 협박을 하다가 무엇으로 현관문을 풀었는지 모르겠는데 문을 열고 들어왔다	d	d
(수사기록 12, 80 쪽)",	d	d
2018. 3.경 범행에 관하여	d	u
"아무 소리 없이 (문을) 따고 들어왔다(수사기록 13, 80 쪽)",	d	d
2018. 7.21.범행에 관하여	d	u
"열쇠를 번호키로 바꾸었는데 어떻게 눌렀는지 번호 누르는 소리가 들리더니 열고 들어왔다(수사기로 14 쪽)"고	d	d

구체적으로 진술하였고,	i	i
"맨 처음(이 사건 각 범죄사실 이전)에는 누구인지 모르는 상태에서 문을 두드려 열어준 사실이 있는데 그다음부터는 문을 열어준 사실이 전혀 없다.	d	d
그 사람이 강제로 문을 따고 들어왔다(수사기록 13 쪽)"고 거듭 강조하였다.	d	d
2 피해자 옆집에 거주하는 D 는 "새벽에 옆집에서 어떤 남자가 문을 세게 치는 소리가 들려 잠에서 깼다.	d	d
원래 번호 키를 누르면 딩동댕 하면서 열려야 하는데 문이 열리지 않아서 '삑삑삑'하는 소리가 2 회 이상 들렸다.	d	d
이게 처음이 아니라 이전에도 한 달에 한 번꼴로 밤 12 시나 새벽에 문을 두드리고 간 적이 있었다"고 진술하였다	d	d
(수사기록 66 쪽).	d	d
3 열쇠집을 운영하는 E 는 "2018. 3. 말경에서 2018. 4. 초순경 피해자가 '어떤 남자가 집에 침입하고 성폭행을 하려고 한다'며 현관문 시정장치를 바꾸어달라고 하여 피해자 집 현관문 시정장치를 번호키 도어락으로 교체해주었다.	d	d
피해자가 집 뒤에 있는 방범창도 봐달라고 하여 확인해 보니,	d	d
방범창이 오래되어 고정이 되어 있지 않아잡아당기면 사람이 들어갈 수 있을 만큼 벌어지는 것을 확인했다	d	i
그래서 전동 드릴과 콘크리트 못으로 피해자 집 방범창을 단단히 고정해주었다"고 진술하였다	d	d
(수사기록 73 쪽).	d	d

2) Argument Relation Classification

Phrase	Related phrase	Golden label	Prediction
피고인은 이 법정에서 범행을 모두 인정하면서도 한편으로 반성문을 통하여 '피해자가 문을 열어주어서 들어갔을 뿐, 피해자의 주거에 침입하지는 아니하였다'는 취지로 주장한다	피고인이 피해자 주거지의 현관문을 불상의 방법으로 열거나 번호키를 해제하는 방법으로 피해자의 주거에 침입한 사실을 충분히 인정할 수 있으므로	a	p
살피건대, 아래와 같은 피해자 및 관련자들의 진술에 의하면,	none	no-rel	no-rel
피고인이 피해자 주거지의 현관문을 불상의 방법으로 열거나 번호키를 해제하는 방법으로 피해자의 주거에 침입한 사실을 충분히 인정할 수 있으므로	피고인의 주장은 받아들이기 어렵다.	s	s
피고인의 주장은 받아들이기 어렵다.	none	no-rel	no-rel
1 피해자는 수사기관에서 2018. 2.경 범행에 관하여	구체적으로 진술하였고,	s	s
"누가 현관문을 두드리더니,뒷창문으로 (집 안을 들여다) 보면서 문 뜯고 들어가겠다고 협박을 하다가 무엇으로 현관문을 풀었는지 모르겠는데 문을 열고 들어왔다	1 피해자는 수사기관에서 2018. 2.경 범행에 관하여	p	p
(수사기록 12, 80 쪽)",	"누가 현관문을 두드리더니,뒷창문으로 (집 안을 들여다) 보면서	p	p

	문 뜯고 들어가겠다고 협박을 하다가 무엇으로 현관문을 풀었는지 모르겠는데 문을 열고 들어왔다		
2018. 3.경 범행에 관하여	(수사기록 12, 80 쪽)",	p	p
"아무 소리 없이 (문을) 따고 들어왔다(수사기록 13, 80 쪽)",	2018. 3.경 범행에 관하여	p	p
2018. 7.21.범행에 관하여	"아무 소리 없이 (문을) 따고 들어왔다(수사기록 13, 80 쪽)",	p	p
"열쇠를 번호키로 바꾸었는데 어떻게 눌렀는지 번호 누르는 소리가 들리더니 열고 들어왔다(수사기로 14 쪽)"고	2018. 7.21.범행에 관하여	p	p
구체적으로 진술하였고,	피고인이 피해자 주거지의 현관문을 불상의 방법으로 열거나 번호키를 해제하는 방법으로 피해자의 주거에 침입한 사실을 충분히 인정할 수 있으므로	s	s
"맨 처음(이 사건 각 범죄사실 이전)에는 누구인지 모르는 상태에서 문을 두드려 열어준 사실이 있는데 그다음부터는 문을 열어준 사실이 전혀 없다.	피고인이 피해자 주거지의 현관문을 불상의 방법으로 열거나 번호키를 해제하는 방법으로 피해자의 주거에 침입한 사실을 충분히 인정할 수 있으므로	s	s
그 사람이 강제로 문을 따고 들어왔다(수사기록 13 쪽)"고 거듭 강조하였다.	"맨 처음(이 사건 각 범죄사실 이전)에는 누구인지 모르는 상태에서 문을 두드려 열어준 사실이 있는데	p	p

	그다음부터는 문을 열어준 사실이 전혀 없다.		
2 피해자 옆집에 거주하는 D는 "새벽에 옆집에서 어떤 남자가 문을 세게 치는 소리가 들려 잠에서 깼다.	피고인이 피해자 주거지의 현관문을 불상의 방법으로 열거나 번호키를 해제하는 방법으로 피해자의 주거에 침입한 사실을 충분히 인정할 수 있으므로	p	p
원래 번호 키를 누르면 딩동댕 하면서 열려야 하는데 문이 열리지 않아서 '삑삑삑'하는 소리가 2 회 이상 들렸다.	2 피해자 옆집에 거주하는 D는 "새벽에 옆집에서 어떤 남자가 문을 세게 치는 소리가 들려 잠에서 깼다.	p	p
이게 처음이 아니라 이전에도 한 달에 한 번꼴로 밤 12 시나 새벽에 문을 두드리고 간 적이 있었다"고 진술하였다	원래 번호 키를 누르면 딩동댕 하면서 열려야 하는데 문이 열리지 않아서 '삑삑삑'하는 소리가 2 회 이상 들렸다.	p	p
(수사기록 66 쪽).	이게 처음이 아니라 이전에도 한 달에 한 번꼴로 밤 12 시나 새벽에 문을 두드리고 간 적이 있었다"고 진술하였다	p	p
3 열쇠집을 운영하는 E는 "2018. 3. 말경에서 2018. 4. 초순경 피해자가 '어떤 남자가 집에 침입하고 성폭행을 하려고 한다'며 현관문 시정장치를 바꾸어달라고 하여 피해자 집 현관문 시정장치를 번호키	피고인이 피해자 주거지의 현관문을 불상의 방법으로 열거나 번호키를 해제하는 방법으로 피해자의 주거에 침입한 사실을 충분히 인정할 수 있으므로	s	p

도어락으로 교체해주었다.			
피해자가 집 뒤에 있는 방범창도 봐달라고 하여 확인해 보니,	3 열쇠집을 운영하는 E 는 "2018. 3. 말경에서 2018. 4. 초순경 피해자가 '어떤 남자가 집에 침입하고 성폭행을 하려고 한다'며 현관문 시정장치를 바꾸어달라고 하여 피해자 집 현관문 시정장치를 번호키 도어락으로 교체해주었다.	p	p
방범창이 오래되어 고정이 되어 있지 않아잡아당기면 사람이 들어갈 수 있을 만큼 벌어지는 것을 확인했다.	피해자가 집 뒤에 있는 방범창도 봐달라고 하여 확인해 보니,	p	s
그래서 전동 드릴과 콘크리트 못으로 피해자 집 방범창을 단단히 고정해주었다"고 진술하였다	방범창이 오래되어 고정이 되어 있지 않아잡아당기면 사람이 들어갈 수 있을 만큼 벌어지는 것을 확인했다.	p	p
(수사기록 73 쪽).	그래서 전동 드릴과 콘크리트 못으로 피해자 집 방범창을 단단히 고정해주었다"고 진술하였다	p	p

3) Argument Structure Visualization