GReX: A Graph Neural Network-Based Rerank-then-Expand Method for Detecting Conflicts Among Legal Articles in Korean Criminal Law

Seonho An[1,2], **Yeong-Yik Rhim**[1,3], **Min-Soo Kim**[1,2,*]
[1]KAIST, Republic of Korea, [2]Infolab, Republic of Korea, [3]Intellicon
{asho1, rhims, minsoo.k}@kaist.ac.kr

Abstract

As social systems become more complex, legal articles have grown increasingly intricate, making it harder for humans to identify potential conflicts among them, particularly when drafting new laws or applying existing ones. Despite its importance, no method has been proposed to detect such conflicts. We introduce a new legal NLP task, *Legal Article Conflict Detection* (LACD), which aims to identify conflicting articles within a given body of law. To address this task, we propose GReX, a novel graph neural network-based retrieval method. Experimental results show that GReX significantly outperforms existing methods, achieving improvements of 44.8% in nDCG@50, 32.8% in Recall@50, and 39.8% in Retrieval F1@50. Our codes are in github.com/asmath472/LACD-public.

1 Introduction

In many countries, courts judge legal cases based on national laws, and lawyers frequently rely on legal articles (also known as *codes*, or *statutes*) in their works. In legal NLP, several studies have utilized legal articles to address tasks such as Legal Judgment Prediction (Feng et al., 2022a,b; Deng et al., 2023; Liu et al., 2023), Legal Article Retrieval (Louis and Spanakis, 2022; Paul et al., 2022; Louis et al., 2023), and Legal Question Answering (Holzenberger et al., 2020; Louis et al., 2024).

Despite their crucial role, some legal articles *conflict* (also known as *contradict*, or *compete*) with one another (Yoon, 2005; Kim, 2005; Araszkiewicz et al., 2021). Here, conflict refers to situations in which overlapping directives or contradictory interpretations arise. For example, in Figure 1, Article 60 of Narcotics Control Act and Article 201 of Criminal Act define different punishments for the same crime, *using opium or morphine*, and thus conflict with each other.

If two articles conflict in a given circumstance, one may be disregarded during judgment, leading to confusion in the application of the law (Yoon, 2005). Detecting such conflicts is therefore essential for individuals involved in drafting laws (e.g., legislators) or enforcing laws (e.g., public prosecutors). As laws grow more complex (Coupette et al., 2021), manually identifying conflicting articles becomes increasingly challenging. Moreover, with the rise of LLMs and agents that rely on natural language rules (Bai et al., 2022; Hua et al., 2024; Dong et al., 2024), automating conflict detection has become even more critical. This study addresses this issue by developing NLP-based methods to automatically detect conflicting articles, with a particular focus on the Criminal Law of the Republic of Korea (hereafter referred to as *Korean Law*).

We introduce a new legal NLP task, **Legal Article Conflict Detection (LACD)**, which aims to *retrieve* articles that conflict with a given *query* article from a collection of legal articles. For example, as illustrated in Figure 1, when Criminal Act Article 201 is given as a query, the model is expected to identify and retrieve Criminal Act Article 205 and Narcotics Control Act Article 60 as conflicting articles, while correctly excluding Narcotics Control Act Article 58-2.

For document retrieval tasks, various methods such as TF-IDF, BM25 (Robertson et al., 2009), DPR (Karpukhin et al., 2020), and retrieve-then-rerank have been widely used. In particular, the *retrieve-then-rerank* approach, where top-ranked candidate documents are reranked using another slower but more accurate language models (LMs), has demonstrated high performance with low latency across various retrieval tasks (Wu et al., 2020; Zhu et al., 2023). However, conventional retrieve-then-rerank methods perform poorly on the LACD task, primarily due to two fundamental differences

*Corresponding author.

Proceedings of the Natural Legal Language Processing Workshop 2025, pages 408–423

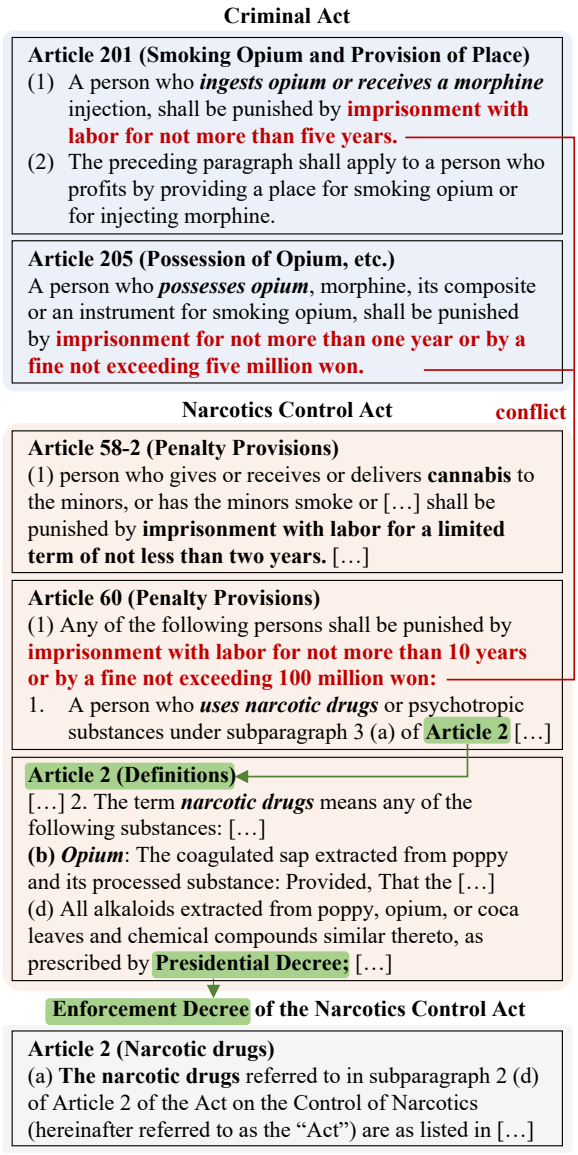

Figure 1: Example of conflicting legal articles in Republic of Korea, translated from Korean. Criminal Act Article 201, Article 205, and Narcotics Control Act Article 60 conflict with one another. In Narcotics Control Act Article 60, ***uses narcotic drugs*** includes ***ingests opium***, in Article 2 of the same act.

(1) between legal documents and general texts, and (2) between LACD and standard retrieval tasks. These differences give rise to two key challenges.

The first challenge (**Challenge 1**) arises from the high textual similarity among legal articles, which hinders the accurate retrieval of conflicting articles in the LACD task (Xu et al., 2020; Paul et al., 2024). For example, in Figure 1, Article 60 and Article 58-2 share nearly identical wording, differing only in their objects (e.g., *narcotic drugs* and *cannibas*) and punishments (e.g., imprisonment for *five years* and *two years*). As a result, an LM may struggles to retrieve Article 60 selectively while filtering out Article 58-2.

The second challenge (**Challenge 2**) lies in the insufficiency of textual descriptions in legal articles, particularly when interpreting legal terminology. Legal articles often rely on references to other articles to define specific terms or conditions (Bommarito II and Katz, 2010; Katz et al., 2020). For instance, in Figure 1, Article 60 uses the term *narcotic drugs*, which is explicitly defined in a referenced article (i.e., *mentioned article*), Article 2. Moreover, accurate interpretation often requires traversing not only direct (i.e., 1-hop) references but also indirect (n-hop) ones. For example, fully understanding the term *narcotic drugs* may require consulting the Enforcement Decree of the Narcotics Control Act. Therefore, to reason effectively over legal articles, a retrieval model must leverage not only the textual content of individual articles but also their explicit inter-article references (i.e., *mention relationships*) (Katz et al., 2020). However, to the best of our knowledge, no prior work has explored the use of mention relationships for legal article retrieval.

To tackle the two key challenges in the LACD task, we propose a novel retrieve-then-rerank method, **GReX**, which consists of two main components: (1) **ReX** (Rerank-then-eXpand), designed to address Challenge 1, and (2) **LGNN** (Legal Graph Neural Network), a reranker aimed to address Challenge 2.

To tackle Challenge 1, the ReX method expands the set of candidate articles using a reranker. It first reranks the initially retrieved top-ranked articles to identify those that conflict with the query article a_q, and then augments the candidate set by including additional articles — originally outside the top-ranked set — that are known to conflict with the identified ones. The underlying intuition builds on transitivity-like relationships among articles observed in prior studies (Bommarito II and Katz, 2010; Boulet et al., 2010; Katz et al., 2020; Coupette et al., 2021), which we extend to conflict relationship: if a_q conflicts with a_j, and a_i is known to conflict with a_j, then a_i is *likely* to conflict with a_q, since such conflicts often reflect overlapping or contradictory legal directives. For example, in Figure 1, if the reranker detects a conflict between Article 201 (the query) and Article 205, and Article 205 is known to conflict with Article 60, ReX expands the candidate set to include Article 60, thereby uncovering its conflict with Article 201.

To tackle Challenge 2, we construct a Legal

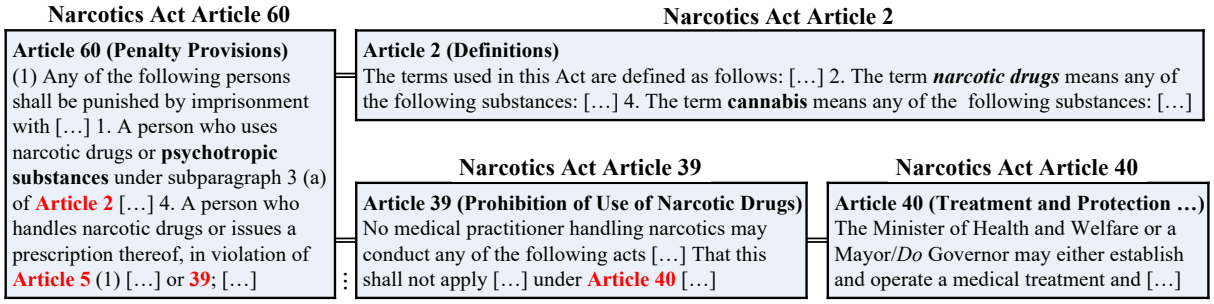

Figure 2: Example of LMGraph (blue box: article, red text: mention). All contents are translated from Korean.

Article Mention Graph (**LMGraph**), where each node represents a legal article and edges represent mention relationships between articles. The LGNN reranker applies a Graph Neural Network (GNN) over this graph to leverage these relationships during reranking. This structure enables contextual reasoning based on inter-article connections. Constructed from Korean law, LMGraph consists of 192,974 nodes and 339,666 edges. Figure 2 shows a small portion of LMGraph. By incorporating this graph, the LGNN reranker gains a deeper understanding of each article within its broader legal context.

We constructed a dedicated dataset for training and evaluating the LACD task, consisting of 392 conflicting article pairs and 3,782 non-conflicting pairs, carefully reviewed and validated by legal experts. We will release it *publicly*. Our proposed retriever, GReX, achieves significant improvements over existing retrieve-then-rerank methods, with improvements of 44.8% in nDCG@50, 32.8% in Recall@50, and 39.8% in Retrieval F1@50.

2 Preliminaries

2.1 Definitions

We define the terms including *case*, *rule*, *article*, *conflict*, and *mention*, largely based on the definitions provided by Araszkiewicz et al. (Araszkiewicz et al., 2021). We use a *legal article* and *article* interchangeably.

Definition 1 (Case, Rule, and Article). A **case** c is a sentence describing the facts of an event (Shao et al., 2020; Sun et al., 2023). A *proposition* (denoted as x) for a case represents an implicit question about its facts. A **rule** r is the implicit legal unit, consisting of a set of propositions (denoted as $\mathcal{X}$) and a judgment p for cases $\mathcal{C}$. A rule r judges a case $c \in \mathcal{C}$ as p if and only if all propositions in $\mathcal{X}$ hold true in c. We denote a rule with $\mathcal{X}$ and p as $r = \mathit{rule}(p, \mathcal{X})$. An **article** a is an explicit legal unit, denoted as $r_i \sqsubseteq a$, and is expressed in sentences.

Definition 2 (Conflict).

1. Two rules conflict, i.e., $\mathit{conflict}(\mathit{rule}(p_1, \mathcal{X}_1), \mathit{rule}(p_2, \mathcal{X}_2))$ if and only if $p_1 \neq p_2$, and $\mathcal{X}_1$ includes $\mathcal{X}_2$, or vice versa.

2. If two rules conflict, then the articles containing those rules also conflict. Specifically, $\mathit{conflict}(a_1, a_2)$ if $\mathit{conflict}(r_i, r_j)$, $r_i \sqsubseteq a_1$, and $r_j \sqsubseteq a_2$.

Definition 3 (Mention). If an article a_1 explicitly cites another article a_2, then a_1 *mentions* a_2.

Most articles implicitly contains at least one rule, making them suitable for the LACD task. Articles that do not contain any rules are discussed in Appendix A.5.4. Examples corresponding to Definitions 1-3 are provided in Appendix A.1.

2.2 Conventional Retrieve-then-Rerank

Given a query article a_q, conventional retrieve-then-rerank methods (Nogueira and Cho, 2019; Wu et al., 2020; Glass et al., 2022; Zhu et al., 2023; Song et al., 2024) retrieve a set of articles $\mathcal{A}_{ret}$ through the following three steps.

1. $\mathbf{v}_{a_q} = \mathit{enc\text{-}bi}(a_q), \mathbf{v}_a = \mathit{enc\text{-}bi}(a)(a \in \mathcal{A})$
2. $\mathcal{A}_{topk} = \{a \mid \text{top-k by } \mathit{sim}(\mathbf{v}_{a_q}, \mathbf{v}_a)\}$
3. $\mathcal{A}_{ret} = \{a_i \mid \text{sort by } \mathit{prob}(\mathit{enc\text{-}cross}(a_q \oplus a_i))\}$ $(a_i \in \mathcal{A}_{topk})$

Here, $\mathbf{v}_a$ is the vector representation of article a; *sim* presents a similarity function, such as inner product; *prob* refers to a layer for calculating retrieval probability; $\oplus$ denotes a textual concatenation operator.

In Step 1, each article $a \in \mathcal{A}$ is pre-encoded into a vector representation $\mathbf{v}_a$ using a bi-encoder, which also encodes the query article a_q into $\mathbf{v}_{a_q}$. In Step 2, the retriever (typically fast) selects the top-k articles $\mathcal{A}_{topk}$ based on the similarity function $\mathit{sim}(\mathbf{v}_{a_q}, \mathbf{v}_a)$. In Step 3, a reranker (typically

slower but more accurate) computes the relevance probability of each article $a_i \in \mathcal{A}_{topk}$ given a_q, using a cross encoder (*enc-cross*) followed by a *prob* layer, and returns the final list $\mathcal{A}_{ret}$ sorted by these probabilities.

3 Methodology

3.1 The LACD task

We define LACD as a retrieval task that takes three inputs: a query article a_q, a collection of articles $\mathcal{A}$, and a set of previously *known* conflicting article pairs $\mathcal{C} = \{(a_i, a_j) \mid \textit{conflict}(a_i, a_j) \wedge a_i \in \mathcal{A} \wedge a_j \in \mathcal{A}\}$. Given these inputs, the task aims to retrieve the set of *unknown* conflicting articles, $\{a_k | \textit{conflict}(a_q, a_k) \wedge a_k \in \mathcal{A} \wedge a_k \notin \mathcal{C}\}$. LACD differs from conventional retrieval tasks in that it focuses on conflictness instead of relevance, and assumes the existence of observed conflicts $\mathcal{C}$.

We refer to the conventional retrieve-then-rerank methods in Section 2.2 as the *Re2* retriever. For the LACD task, the Re2 retriever often fails to accurately identify conflicting articles. Figure 3 shows an example of applying Re2 to LACD. While Re2 correctly retrieves Article 205 (a_1), it fails to retrieve Article 60 (a_2) since a_2 is not included in $\mathcal{A}_{topk}$, even though it potentially conflicts with a_q. This occurs because Re2 prioritizes *semantically irrelevant* articles such as a_3 over a_2 (Challenge 1). Even when $\mathcal{A}_{topk}$ includes a_2 by increasing k, Re2 still struggles to detect the conflict in Step 3 due to its reliance on referenced definitions in mentioned articles (Challenge 2).

3.2 The GReX method

The GReX method reformulates Steps 2 and 3 of the Re2 method, as illustrated in Figure 4. Step 2 is enhanced by the proposed ReX method, which incorporates external conflicting articles when constructing $\mathcal{A}_{topk}$. Step 3 is improved by integrating GNN-based embeddings from LMGraph into the LGNN reranker, thereby refining the final retrieval set $\mathcal{A}_{ret}$. We describe the core components of GReX in detail: ReX, LMGraph, and the LGNN reranker.

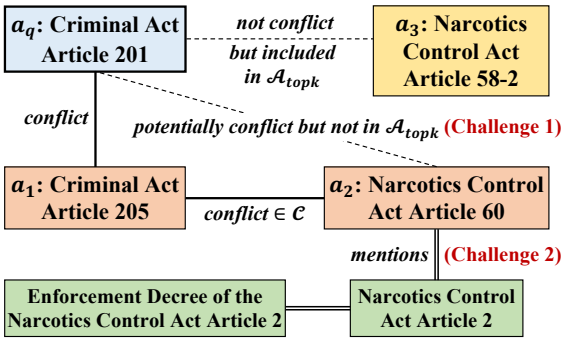

Figure 3: An example of applying Re2 to LACD.

Rerank-then-eXpand (ReX): The ReX method enhances $\mathcal{A}_{topk}$ by selectively expanding it. Specifically, (1) ReX first identifies articles that directly conflict with the query article using the reranker (e.g., a_1 in Figure 3), and (2) expands $\mathcal{A}_{topk}$ by including articles known to conflict with those identified, such as a_2, where $(a_1, a_2) \in \mathcal{C}$. This approach leverages the *triadic closure* phenomenon frequently observed in legal article conflicts, where descriptions among conflicting articles often exhibit significant overlap. For example, Article 201, Article 205, and Article 60 in Figure 1 and 3, all describe crimes involving opium, creating an overlap and thus forming a triadic closure. We further show that triadic closure is *guaranteed* under specific conditions, as detailed in Appendix A.2. The ReX method performs the following three substeps for Step 2.

- **Step 2-1:** Retrieve $\mathcal{A}_{topk}$ (same as in Re2).
- **Step 2-2:** Rerank $\mathcal{A}_{topk}$ and select a subset $\mathcal{A}_{filter} \subseteq \mathcal{A}_{topk}$, defined as $\mathcal{A}_{filter} = \{a_i \in \mathcal{A}_{topk} \mid \textit{prob}(\textit{enc-cross}(a_q \oplus a_i)) > \theta\}$.
- **Step 2-3:** Expand $\mathcal{A}_{topk}$ by augmenting it with articles $a \in \mathcal{A}$ such that there exists $a_i \in \mathcal{A}_{filter}$ with $(a_i, a) \in \mathcal{C}$. That is, $\mathcal{A}_{topk} \leftarrow \mathcal{A}_{topk} \cup \{a \in \mathcal{A} \mid \exists a_i \in \mathcal{A}_{filter}, (a_i, a) \in \mathcal{C}\}$.

Here, we use $\min/P_{TC}$ as the default threshold θ, where min denotes the minimum of $\{\textit{rerank}(a_q, a_i) \mid a_i \in \mathcal{A}_{topk}\}$ and P_{TC} is the conditional probability of triadic closure, defined as:

$$P_{TC} = P((a_i, a_k) \in \mathcal{C} \mid (a_i, a_j), (a_j, a_k) \in \mathcal{C})$$

A justification for Step 2-2 is in Appendix A.3.

In Step 2-2, the reranker has already been fine-tuned using $\mathcal{C}$; however, $\mathcal{C}$ is used again in a non-parametric manner to further enhance performance. In Figure 4, the yellow node (article) represents $\mathcal{A}_{filter}$. In Step 2-3, while most articles in $\mathcal{A} \backslash \mathcal{A}_{topk}$ may not conflict with a_q, the augmented articles are likely to do so, as they explicitly conflict with articles in $\mathcal{A}_{filter}$. A formal proof for Step 2-3 is in Appendix A.2. In Figure 4, the red node represents the augmented articles.

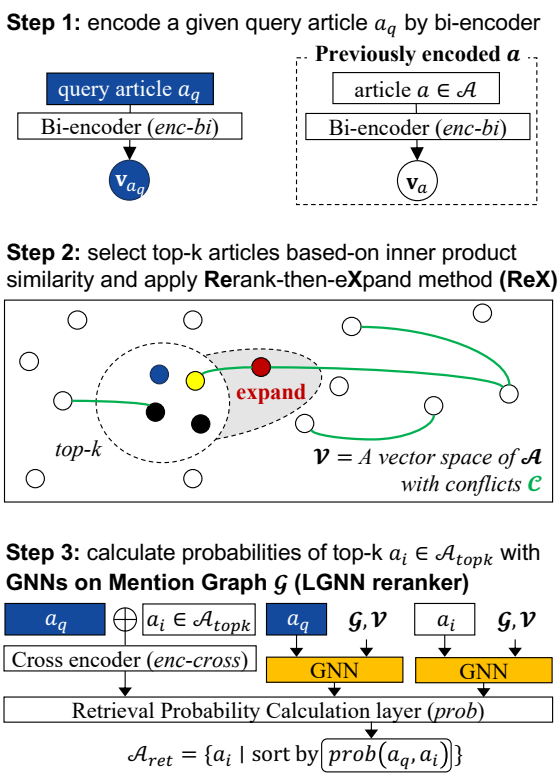

Figure 4: Outline of GReX. The blue, yellow, and black circles represent the query article, an article with a high rerank score, and articles with low rerank scores, respectively.

LMGraph: We construct LMGraph $\mathcal{G}$, where nodes represent legal articles $\mathcal{A}$, and edges $\mathcal{E}$ represent the *mention relationships* among these articles. The mention relationships are identified based on specific textual templates, such as '제∥num 조' (meaning 'Article *num*'), following the guidelines in (Ministry of Government Legislation, 2023). Formally, the edge set $\mathcal{E}$ is defined as follows:

$$\mathcal{E} = \{(a_i, a_j) \mid a_i \text{ mentions } a_j \text{ or } a_j \text{ mentions } a_i\}$$

All articles and mention relationships are based on a snapshot taken on September 30, 2024, and were obtained via crawling the Ministry of Government Legislation website[1]. Copyright considerations related to this process are discussed in Section 7. As a result, the constructed LMGraph for Korean law consists of 192,974 nodes and 339,666 edges, with detailed statistics presented in Table 1.

Statistics	Avg. value	Std. deviation
# of words per article	75.5	90.6
# of edges per node	4.57	11.11

Table 1: Statistics about LMGraph for Korean Law.

[1]Ministry of Government Legislation official site

LGNN reranker: The LGNN reranker combines the output of the cross-encoder with node representations from the GNN encoder, which captures not only text-level conflicts between articles but also semantic relationships through multi-hop connections in LMGraph. This enables the model to reject articles that are textually conflicting with the query but semantically irrelevant. Specifically, for each pair (a_q, a_i), we concatenate the output of the cross-encoder, *enc-cross*$(\cdot)$, with the similarity score *sim*$(\cdot)$ computed between the node representations of a_q and a_i. Formally, it enhances Step 3 for the LACD task as follows:

Step 3. $\mathcal{A}_{ret} = \{a_i \mid \text{sort by } prob\big(enc\text{-}cross(a_q \oplus a_i) || sim(GNN(\mathcal{G}, \mathcal{V})_q, GNN(\mathcal{G}, \mathcal{V})_i)\big)\}(a_i \in \mathcal{A}_{topk})$

where $||$ denotes the concatenation operator, and *GNN*$(\cdot)$ represents the output of the GNN encoder. Step 3 computes *enc-cross*$(\cdot)$ only for the articles augmented in Step 2-3 since those in $\mathcal{A}_{topk}$ were already computed in Step 2-2.

GNN$(\cdot)$ takes as input the LMGraph $\mathcal{G} = (\mathcal{A}, \mathcal{E})$, where each article $a_i \in \mathcal{A}$ has its own initial feature representation $h_{a_i}^{(0)}$. In general, the GNN consists of L layers, producing a list of node representations from the initial layer $\mathbf{H}^{(0)} = [h_{a_1}^{(0)}, \cdots, h_{a_{|\mathcal{A}|}}^{(0)}]$ to the final layer $\mathbf{H}^{(L)} = [h_{a_1}^{(L)}, \cdots, h_{a_{|\mathcal{A}|}}^{(L)}]$. In this representation, $GNN(\mathcal{G}, \mathcal{V})_q$ refers to the final feature vector $h_{a_q}^{(L)}$ for node a_q, and similarly, $GNN(\mathcal{G}, \mathcal{V})_i$ refers to $h_{a_i}^{(L)}$. Each GNN layer l updates the node representations $\mathbf{H}^{(l)}$ using the edge weight matrix $\mathbf{A}^{(l)}$ and transformation weights $\mathbf{W}^{(l)}$ as follows:

$$\mathbf{H}^{(l+1)} = \sigma(\mathbf{A}^{(l)}\mathbf{H}^{(l)}\mathbf{W}^{(l)})$$

For our LGNN reranker, we adopt a two-layer GATv2 architecture (Brody et al., 2022) as the default GNN model. In GATv2, the edge weights $\mathbf{A}_i^{(l)}$ for i-th node (i.e., node for a_i) at layer l are given by:

$$\mathbf{A}_i^{(l)} = \sigma\Big(\sum_{j \text{ for} (a_i, a_j) \in \mathcal{E}} \text{softmax}\Big(\sigma(\text{att}(h_{a_i}^{(l)}, h_{a_j}^{(l)}))\Big)\Big)$$

4 Experimental settings

4.1 The LACD dataset

To construct the dataset for the LACD task in Korean Law, we collected 4,174 pairs of articles (a_1, a_2), each manually labeled them as either conflicting or non-conflicting. The criteria used to collect these pairs are summarized in Appendix A.4.

We randomly split these pairs into 60% for training, 20% for validation, and 20% for testing. From the conflicting pairs (a_1, a_2) in the test set, we extract all unique articles (either a_1 or a_2) and use them as query articles (89 in total). We also construct $\mathcal{C}$ (*seen* conflicts) as the set of conflict pairs from the training and validation sets. Since conflict is symmetric, the number of pairs in $\mathcal{C}$ is approximately twice the number of unique conflict pairs. For simplicity, we assume that all articles in corpus for retrieval are drawn from Acts (excluding Enforcement Decrees). Further details on the legal hierarchy are in Appendix A.5.1. Detailed statistics are provided in Table 2.

Quality review: Our dataset was validated by legal experts. As a result, nearly 94% of the pairs aligning with real-world conflict and the remaining 6% differing but still fitting our definitions. Detailed explanations about conflicts in real worlds and quality review questions in Appendix A.5.

Datasets	Conflict	Non-conflict	Avg. # of words
Train	226	2,278	120.89
Validation	90	745	118.97
Test	76	759	124.12
Total	392	3,782	121.15
# of queries	**# of corpus**	**# of unseen conflicts per query**	
89	79,615	1.69	
# of pairs in $\mathcal{C}$	**# of articles**	**Avg. # of conflicts per article**	
630	199	3.17	

Table 2: Statistics for (a_1, a_2) pairs (upper table), the test queries in the LACD dataset (middle table), and the seen conflicts (lower table).

4.2 Baselines

We use KoBigBird (Park and Kim, 2021) as the bi-encoder(*enc-bi*), and Klue/RoBERTa (Park et al., 2021) as the cross-encoder. For vector storage and retrieval, we employ Chroma DB[2], using the inner product as $sim(\cdot)$.

Since Re2 and GReX differ at Steps 2 and 3, there are four possible combinations: (1) **Re2** (conventional method), (2) **Re2+LGNN** (Re2 at Step 2 + LGNN at Step 3), (3) **ReX+Re2** (ReX at Step 2 + Re2 at Step 3), and (4) **GReX** (proposed method). These combinations are evaluated in our experiments, while comparisons with additional baselines are presented in Section 5.3.

Among the above combinations, **ReX+Re2** and **GReX** actually rerank more articles than the others by expanding $\mathcal{A}_{topk}$. For example, when $k = 100$, the former typically reranks approximately 150 articles, while the latter reranks exactly 100. To ensure fair comparisons, we set $k = 150$ for **Re2** and **Re2+LGNN** so that the number of reranked articles is comparable.

[2]Chroma DB official site

4.3 Training and evaluations

We build both Re2 and GReX retrievers using a pre-trained bi-encoder model and a fine-tuned cross encoder. We denote the set of labeled articles pairs use for training (226 + 2278 pairs as shown in Table 2) as $\mathcal{S}$, where each pair $s = (a_i, a_j) \in \mathcal{S}$. Then, the training objective $\hat{y_s}$ is defined as follows, where $rank(a_q, a_i)$ denotes $enc\text{-}cross(a_q \oplus a_i)$ in Re2, and $enc\text{-}cross(a_q \oplus a_i)||sim(GNN(\mathcal{G}, \mathcal{V})_q, GNN(\mathcal{G}, \mathcal{V})_i)$ in GReX.

$$\hat{y_s} = prob(rank(a_q, a_i))$$

For training, we use the *Weighted Binary Cross Entropy loss* as the loss function $\mathcal{L}$, defined as follows.

$$\mathcal{L} = -\frac{1}{N}\sum_{s\in\mathcal{S}} w_T y_s \log(\hat{y_s}) + w_F(1-y_s)\log(1-\hat{y_s})$$

Here, $N = \|\mathcal{S}\|$ denotes the number of training pairs, y_s is the ground-truth label for each pair s, and $0 < w_T, w_F < 1$ are the weights for the true and false labels, respectively. Details of the training and testing are in Appendix A.6.

When evaluating GReX, we exclude $a_i \in \mathcal{A}_{ret}$ such that $(a_q, a_i) \in \mathcal{C}$ to avoid retrieving *seen* conflict pairs. We also evaluate our LGNN reranker on other Korean legal NLP dataset, as detailed in Appendix A.7. As evaluation metrics, we use nDCG@n, recall@n, and retrieval F1@n (definitions are in Appendix A.8). Each experiment is run three times, and we report the mean performance with standard deviation. Significance test results are in Appendix A.9.

5 Results and analysis

5.1 Main results

Table 3 shows the performance of three GReX variants and other baselines for the full retrieval pipeline. GReX significantly outperforms Re2 by 44.8% in nDCG@50, 32.8% in Recall@50, and 39.8% in Retrieval F1@50. We also observe a *synergistic effect* between ReX and LGNN. For example, in nDCG@10, ReX and LGNN individually improve Re2 by 3.24%p and by 3.65%p, respectively, whereas GReX achieves a larger improvement of 10.53%p, exceeding the sum of individual improvements. This indicates that the LGNN

Methods	nDCG@n			Recall@n			Retrieval F1@n		
	n=5	*n=10*	*n=50*	*n=5*	*n=10*	*n=50*	*n=5*	*n=10*	*n=50*
Retrieve									
TF-IDF	15.59	18.31	22.26	25.28	32.43	48.67	11.47	8.75	3.05
BM25	14.14	16.67	19.86	23.50	29.66	44.08	10.48	8.14	2.57
enc-bi	13.68	14.69	18.70	22.75	25.97	41.40	10.40	6.81	2.65
Retrieve-then-rerank									
Re2	16.38±1.48	20.38±1.97	26.09±0.98	24.00±1.65	35.15±3.64	57.71±1.34	12.01±0.81	9.91±0.83	3.67±0.09
Re2+**LGNN**	20.55±2.55	24.03±2.37	29.58±1.85	27.44±4.41	36.86±2.35	59.27±0.40	13.46±2.00	10.47±0.78	3.79±0.00
ReX+Re2	18.02±0.71	23.62±1.13	31.62±0.45	28.37±2.21	43.84±3.14	74.98±0.88	13.99±0.91	12.52±0.61	4.94±0.05
GReX (ours)	**25.27±0.95**	**30.91±1.51**	**37.79±1.00**	**34.88±1.35**	**50.34±2.83**	**76.65±1.05**	**16.82±0.57**	**13.99±0.90**	**5.13±0.07**

Table 3: Performance (%) across all Steps. The best and second results are highlighted in **bold** and underline, respectively. *enc-bi* denotes Re2 without reranking. The four retrieve-and-rerank methods are detailed in Section 4.2.

reranker provides higher-quality scores than the naïve reranker in Re2, which in turn enhances the quality of $\mathcal{A}_{filter}$ selected by ReX.

ReX+Re2 performs worse than Re2+LGNN at smaller values of n (e.g., nDCG@5 and nDCG@10), where accurate reranking is more critical, due to its reliance on Re's naive reranker. In contrast, at larger n (e.g., nDCG@50), where expanding $\mathcal{A}_{topk}$ becomes more important, Re2+LGNN performs worse than ReX+Re2, since Re2 (i.e., Step 2) does not expand $\mathcal{A}_{topk}$. Further details are in the *Error Analysis* in Section 5.4.

5.2 ReX on synthetic $\mathcal{C}$

To evaluate the robustness of ReX, we construct and use a synthetic conflict set $\mathcal{C}_{syn}$, instead of using $\mathcal{C}$. Specifically, we collect all distinct articles $\mathcal{D}$ from the training data $\mathcal{S}$, excluding those used as test query articles, and compute $\mathcal{C}_{syn} = \{(a_i, a_j) \mid a_i \in \mathcal{D} \wedge a_j \in \mathcal{A}_{topk}(a_i) \wedge prob(rank(a_i, a_j)) > 0.5\}$, indicating article pairs with relatively high likelihood of conflict. Here, $\mathcal{A}_{topk}(a_i)$ denotes $\mathcal{A}_{topk}$ retrieved by *enc-bi* using a_i as the query.

Table 4 presents the performance when using $\mathcal{C}_{syn}$. Results for Re2+LGNN are omitted, since it does not use the conflict set. ReX+Re2 yields lower performance than Re2 under $\mathcal{C}_{syn}$, due to a fundamental difference between the LGNN reranker and Re2's naive reranker: the former captures external definitions in articles, whereas the latter does not. In contrast, GReX significantly outperforms both Re2 and ReX+Re2 even when using $\mathcal{C}_{syn}$, owing to the improved quality of $\mathcal{A}_{topk}$ and the synergistic effect between ReX and LGNN described above.

5.3 Other baselines in LACD

To validate the effectiveness of our methods, we additionally compare them against a well-known $\mathcal{A}_{topk}$ refinement method: *Pseudo Relevance Feedback using Rocchio algorithm* (Rocchio-PRF) (Rocchio Jr, 1971; Croft and Harper, 1979; Gao et al., 2023). In this experiment, Rocchio-PRF updates the query vector $\mathbf{v}_{a_q}$ as follows:

$$\mathbf{v}_{a_q} \leftarrow (\mathbf{v}_{a_q} + \sum_{a_i \in \mathcal{A}_{topk}} \mathbf{v}_{a_i})/(k+1)$$

After this update, $\mathcal{A}_{topk}$ is re-retrieved using the new $\mathbf{v}_{a_q}$. Since Rocchio-PRF enhances $\mathcal{A}_{topk}$ (i.e., improve Step 2), there are two possible combinations: (1) Rocchio-PRF+Re2 (Rocchio-PRF at Step 2 and Re2 at Step 3), and (2) Rocchio-PRF+LGNN (Rocchio-PRF at Step 2 and LGNN reranker at Step 3).

Methods	nDCG@n		
	n=5	*n=10*	*n=50*
Re2†	16.38	20.38	26.09
ReX+Re2 using $\mathcal{C}_{syn}$	13.41	18.47	25.49
ReX+Re2 using $\mathcal{C}^\dagger$	18.02	23.62	31.62
GReX using $\mathcal{C}_{syn}$	21.54	26.29	32.16
GReX using $\mathcal{C}^\dagger$	**25.27**	**30.91**	**37.79**

Table 4: Performance comparison of ReX using $\mathcal{C}$ and $\mathcal{C}_{syn}$. † indicates results reported in Table 3.

Table 5 presents the performance of the Rocchio-PRF variants, along with Re2 and our GReX. GReX significantly outperforms both Rocchio-PRF variants. ReX+Re2 ranks second, highlighting the contribution of ReX. Among the Rocchio-PRF variants, Rocchio-PRF+LGNN outperforms Rocchio-PRF+Re2, demonstrating the effectiveness of LGNN not only within the Re2 pipeline but also when applied to the $\mathcal{A}_{topk}$ set refined by Rocchio-PRF.

5.4 Category-wise performance analysis

Categorization of Articles: We categorize all unseen conflict pairs $\{(a_q, q_i)\}$ for all 89 queries (150 $=$ 1.69 $\times$ 89 pairs in total) in Table 2(middle) into four groups: *Criminal* (78 pairs), *Mention* (12 pairs), *Both* (18 pairs), and *Neither* (42

Methods	nDCG@n		
	n=5	*n=10*	*n=50*
Re2†	16.38	20.38	26.09
Rocchio-PRF + Re2	13.45	17.19	21.82
Rocchio-PRF + LGNN	16.51	19.52	23.90
ReX+Re2†	18.02	23.62	31.62
GReX†	**25.27**	**30.91**	**37.79**

Table 5: Performance comparison between Rocchio-PRF and GReX. † indicates results reported in Table 3.

pairs). *Criminal* indicates that both a_q and a_i belong to the *Criminal Act*; *Mention* denotes that there is a mention relationship between a_q and a_i; *Both* refers to pairs that satisfy both conditions; and *Neither* denotes pairs that satisfy neither. This categorization is based on the relevance of the *Criminal Act*, which is the main focus of this study, and the observation that a pair of conflict articles including a mention relationship is relatively difficult to retrieve using *enc-bi*.

Figure 5 shows an example of a conflict pair in the *Mention* category. Article 324-2 provides a self-contained crime description, allowing *enc-bi* to generate an accurate semantic representation. In contrast, Article 324-4 references external definitions in Article 324-2, resulting in an inaccurate semantic representation. Consequently, the semantic similarity between both is low, hindering Re2 from retrieving one given the other as a query.

Criminal Act

Article 324-2 (Coercion by Hostage)
A person who arrests or confines another or obtains or maintains another […] shall be punished by **imprisonment with labor for a limited term of at least three years.**

Article 324-4 (Murder of Hostage)
If a person who has committed the crime as prescribed in **Article 324-2**, murders the hostage, the person shall be punished **by death or imprisonment for an indefinite term.** […]

Figure 5: Example of conflict pairs in *Mention* category.

Analysis by Category: Figure 6 shows Recall@50 results for four methods across the four categories. GReX consistently outperforms Re2, demonstrating its effectiveness across diverse article types. Both Re2 and Re2+LGNN exhibits the lowest performance in the *Mention* category. This is because Re2's naive retriever inherently struggles to retrieve conflicting articles including mention relationships, as explained above, resulting in low-quality $\mathcal{A}_{topk}$.

The result of Re2+LGNN indicates that using the LGNN reranker alone does not improve performance for this category. The LGNN reranker is designed to compensate for contextual deficiencies caused by mention relationships through GNN-based propagation. However, as shown in Figure 6, Article 324-4 mentions Article 324-2, and interpreting the former requires only the content of the latter, which is already provided to *enc-cross* without GNNs. Thus, the benefit of additional propagation is minimal in such cases.

In contrast, Figure 6 show that both ReX+Re2 and GReX achieve their best performance in the *Mention* category among all categories. It is because ReX significantly enhances the quality of $\mathcal{A}_{topk}$ through reranking (Step 2-2), which effectively leverages mention relationships. GReX does not further improve performance over ReX+Re2 for the same reason as Re2+LGNN.

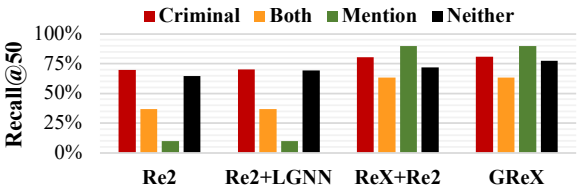

Figure 6: Recall@50 of methods for four categories.

6 Related works

Legal article retrieval: Legal article retrieval, which focuses on finding relevant legal articles given a query, has been extensively studied (Louis and Spanakis, 2022; Paul et al., 2022; Louis et al., 2023; Su et al., 2024; Chen et al., 2025). The retrieved articles are widely used in downstream legal NLP tasks, such as legal QA and judgment prediction (Louis et al., 2024; Qin et al., 2024). Some prior studies have improved retrieval performance by applying GNNs to article hierarchy or article–case graphs (Paul et al., 2022; Louis et al., 2023). However, these methods are not applicable to the conflict detection task.

Korean legal NLP: Recent studies in Korean legal NLP have explored various tasks, such as LJP (Hwang et al., 2022), legal reasoning (Kim et al., 2024a,b), and response evaluation in the legal domain (Ryu et al., 2023). However, no prior work has addressed legal article retrieval or conflict detection, nor has any dataset included mention relationships similar to our LMGraph.

7 Conclusions

In this paper, we proposed a new legal NLP task, Legal Article Conflict Detection (LACD), and constructed a dedicated dataset for it. We propose a novel retriever, GReX, which integrates two key

techniques: ReX and the LGNN reranker. Experimental results demonstrate that GReX significantly outperforms existing retrievers on the LACD task.

Limitations

In this paper, we propose GReX as a solution to address the problem of legal conflict detection. However, our approach has several limitations:

First, our methodology has only been validated within the domain of criminal law in Korea. Korean criminal law is one of the most extensively studied areas related to legal conflict, and it provides a convenient basis for dataset creation. However, it is necessary to expand this research to other domains, such as civil, building or administrative law, to address legal conflict comprehensively in the future.

Second, our LMGraph only incorporates mention relationships between articles as edges. For example, methods like G-DSR (Louis et al., 2023) utilize tree structures within laws as links, which our approach does not include. Whether incorporating such tree structures could effectively solve the LACD problem remains out of scope for this work and requires future investigation.

Lastly, our study focuses exclusively on conflicts between articles that contain one or more rules. Conflicts involving articles without rules (e.g., definitional conflicts) are beyond the scope of this work and remain an open area for future research.

Ethical considerations

Language models have inherent issues with hallucination and the potential to generate biased outputs. In particular, when identifying conflicts, models may incorrectly retrieve relevant legal articles. Furthermore, although the term *conflict* generally carries a negative connotation, this does not imply that a conflicting legal article is *inherently problematic* or *should necessarily be deleted*. Under Korean law, as detailed in Appendix A.5, there are some procedures to resolve such conflicts. Indeed, some articles are explicitly drafted with the potential for conflict in mind.

The mention relationships in law was obtained by crawling data from the official website of the Ministry of Government Legislation. According to Article 7 of the Copyright Act in Korea, legal provisions and compilations of laws created by the government (including link information) are not protected as copyrighted works.

References

Michał Araszkiewicz, Enrico Francesconi, and Tomasz Zurek. 2021. Identification of contradictions in regulation. In *Legal Knowledge and Information Systems*, pages 151–160. IOS Press.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Michael J Bommarito II and Daniel M Katz. 2010. A mathematical approach to the study of the united states code. *Physica A: Statistical Mechanics and its Applications*, 389(19):4195–4200.

Romain Boulet, Pierre Mazzega, and Danièle Bourcier. 2010. Network analysis of the french environmental code. In *AI Approaches to the Complexity of Legal Systems. Complex Systems, the Semantic Web, Ontologies, Argumentation, and Dialogue: International Workshops AICOL-I/IVR-XXIV Beijing, China, September19, 2009 and AICOL-II/JURIX 2009, Rotterdam, The Netherlands, December 16, 2009 Revised Selected Papers*, pages 39–53. Springer.

Shaked Brody, Uri Alon, and Eran Yahav. 2022. How attentive are graph attention networks? In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Zhe Chen, Pengjie Ren, Fuhui Sun, Xiaoyan Wang, Yujun Li, Siwen Zhao, and Tengyi Yang. 2025. SLARD: A Chinese superior legal article retrieval dataset. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 740–754, Abu Dhabi, UAE. Association for Computational Linguistics.

Corinna Coupette, Janis Beckedorf, Dirk Hartung, Michael Bommarito, and Daniel Martin Katz. 2021. Measuring law over time: A network analytical framework with an application to statutes and regulations in the united states and germany. *Frontiers in Physics*, 9:658463.

WB Croft and DJ Harper. 1979. Using probabilistic models of document retrieval without relevance information. volume 35, pages 285–295. MCB UP Ltd.

Wentao Deng, Jiahuan Pei, Keyi Kong, Zhe Chen, Furu Wei, Yujun Li, Zhaochun Ren, Zhumin Chen, and Pengjie Ren. 2023. Syllogistic reasoning for legal judgment analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13997–14009, Singapore. Association for Computational Linguistics.

Yi Dong, Ronghui Mu, Gaojie Jin, Yi Qi, Jinwei Hu, Xingyu Zhao, Jie Meng, Wenjie Ruan, and Xiaowei Huang. 2024. Position: Building guardrails for

large language models requires systematic design. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Yi Feng, Chuanyi Li, and Vincent Ng. 2022a. Legal judgment prediction: A survey of the state of the art. In *IJCAI*, pages 5461–5469.

Yi Feng, Chuanyi Li, and Vincent Ng. 2022b. Legal judgment prediction via event extraction with constraints. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 648–664, Dublin, Ireland. Association for Computational Linguistics.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2020. A dataset for statutory reasoning in tax law entailment and question answering. In *NLLP@KDD*.

Wenyue Hua, Xianjun Yang, Mingyu Jin, Zelong Li, Wei Cheng, Ruixiang Tang, and Yongfeng Zhang. 2024. Trustagent: Towards safe and trustworthy llm-based agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10000–10016.

Wonseok Hwang, Dongjun Lee, Kyoungyeon Cho, Hanuhl Lee, and Minjoon Seo. 2022. A multi-task benchmark for korean legal language understanding and judgement prediction. In *Advances in Neural Information Processing Systems*, volume 35, pages 32537–32551. Curran Associates, Inc.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Daniel Martin Katz, Corinna Coupette, Janis Beckedorf, and Dirk Hartung. 2020. Complex societies and the growth of the law. *Scientific reports*, 10(1):18737.

Minju Kim, Haein Jung, and Myoung-Wan Koo. 2024a. Self-expertise: Knowledge-based instruction dataset augmentation for a legal expert language model. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1098–1112.

Seong-Don Kim. 2005. Fallgruppen der gesetzeskonkurrenz und ihre bewertungsmethode. *Korean Lawyers Association Journal*, 54(1):29–67.

Yeeun Kim, Youngrok Choi, Eunkyung Choi, JinHwan Choi, Hai Jin Park, and Wonseok Hwang. 2024b. Developing a pragmatic benchmark for assessing Korean legal language understanding in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5573–5595, Miami, Florida, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yifei Liu, Yiquan Wu, Yating Zhang, Changlong Sun, Weiming Lu, Fei Wu, and Kun Kuang. 2023. Mlljp: multi-law aware legal judgment prediction. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 1023–1034.

Antoine Louis and Gerasimos Spanakis. 2022. A statutory article retrieval dataset in French. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6789–6803, Dublin, Ireland. Association for Computational Linguistics.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2023. Finding the law: Enhancing statutory article retrieval via graph neural networks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2761–2776, Dubrovnik, Croatia. Association for Computational Linguistics.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. Interpretable long-form legal question answering with retrieval-augmented large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22266–22275.

Ministry of Government Legislation. 2023. Standards for legislative drafting and review. Accessed: 2024-12-15.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Jangwon Park and Donggyu Kim. 2021. Kobigbird: Pretrained bigbird model for korean.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Taehwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong,

Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Lucy Park, Alice Oh, Jungwoo Ha, and Kyunghyun Cho. 2021. Klue: Korean language understanding evaluation.

Shounak Paul, Rajas Bhatt, Pawan Goyal, and Saptarshi Ghosh. 2024. Legal statute identification: A case study using state-of-the-art datasets and methods. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2231–2240, New York, NY, USA. Association for Computing Machinery.

Shounak Paul, Pawan Goyal, and Saptarshi Ghosh. 2022. Lesicin: A heterogeneous graph-based approach for automatic legal statute identification from indian legal documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11139–11146.

Weicong Qin, Zelin Cao, Weijie Yu, Zihua Si, Sirui Chen, and Jun Xu. 2024. Explicitly integrating judgment prediction with legal document retrieval: A law-guided generative approach. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2210–2220, New York, NY, USA. Association for Computing Machinery.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Joseph John Rocchio Jr. 1971. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*.

Cheol Ryu, Seolhwa Lee, Subeen Pang, Chanyeol Choi, Hojun Choi, Myeonggee Min, and Jy-Yong Sohn. 2023. Retrieval-based evaluation for LLMs: A case study in Korean legal QA. In *Proceedings of the Natural Legal Language Processing Workshop 2023*, pages 132–137, Singapore. Association for Computational Linguistics.

Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. 2020. Bert-pli: Modeling paragraph-level interactions for legal case retrieval. In *IJCAI*, pages 3501–3507.

EuiYul Song, Sangryul Kim, Haeju Lee, Joonkee Kim, and James Thorne. 2024. Re3val: Reinforced and reranked generative retrieval. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 393–409, St. Julian's, Malta. Association for Computational Linguistics.

Weihang Su, Yiran Hu, Anzhe Xie, Qingyao Ai, Quezi Bing, Ning Zheng, Yun Liu, Weixing Shen, and Yiqun Liu. 2024. STARD: A Chinese statute retrieval dataset derived from real-life queries by non-professionals. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10658–10671, Miami, Florida, USA. Association for Computational Linguistics.

Zhongxiang Sun, Jun Xu, Xiao Zhang, Zhenhua Dong, and Ji-Rong Wen. 2023. Law article-enhanced legal case matching: A causal learning approach. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1549–1558.

Julián Urbano, Harlley Lima, and Alan Hanjalic. 2019. Statistical significance testing in information retrieval: an empirical analysis of type i, type ii and type iii errors. In *Proceedings of the 42nd International ACM SIGIR conference on Research and development in information retrieval*, pages 505–514.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish confusing law articles for legal judgment prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3086–3095, Online. Association for Computational Linguistics.

Dong-Ho Yoon. 2005. Grundforschung zur reform der sonderstrafgesetzbuche. *Korean Institute of Criminology and Justice*, pages 9–282.

Fangwei Zhu, Jifan Yu, Hailong Jin, Lei Hou, Juanzi Li, and Zhifang Sui. 2023. Learn to not link: Exploring NIL prediction in entity linking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10846–10860, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Examples for article conflict

In this section, we provide some examples which well explain our definitions in Section 2.1.

Example 1. We can represent Article 205 in Figure 1 and its example case c_1 as follows:

$$c_1 = \text{Bob smoked opium in his house.}$$

$$a_1 = \begin{array}{l}\texttt{Criminal Act Article 205 (Posses-}\\ \texttt{sion of Opium, ... million won.}\end{array}$$

$$\mathcal{X}_1 = \left\{\begin{array}{l}\text{Is a person possesses } \textit{something}?\\ \text{Is } \textit{something} \in \text{opium} \vee \text{morphine} \cdots ?\end{array}\right\}$$

$$p_1 = \text{Less than five million won fine}$$
$$\vee \text{ Less than one year imprisonment.}$$

$$r_1 = \textit{rule}(p_1, \mathcal{X}_1),\ r_1 \sqsubseteq a_1$$

Since Bob *smoked* (a proposition about possession in $\mathcal{X}_1$) *opium* (a proposition regarding opium $\vee$ morphine $\vee \cdots$ in $\mathcal{X}_1$), all propositions in $\mathcal{X}_1$ hold in c_1, and thus, case c_1 is judged as p_1.

Example 2. We can represent Article 201 in Figure 1 as follows:

$$a_2 = \begin{array}{l}\texttt{Criminal Act Article 201 (Smoking}\\ \texttt{Opium and Provision ... morphine.}\end{array}$$

$$\mathcal{X}_2 = \left\{\begin{array}{l}\text{Is a person uses } \textit{something}?\\ \text{Is } \textit{something} \in \text{opium} \vee \text{morphine}?\end{array}\right\}$$

$$p_2 = \text{Labor not more than five years.}$$

$$r_2 = \textit{rule}(p_2, \mathcal{X}_2),\ r_2 \sqsubseteq a_2$$

Here, *possesses* $\mathcal{X}_1$ includes *uses* $\mathcal{X}_2$, and *opium*$\vee$ *morphine* $\cdots$ $\mathcal{X}_1$ includes *opium* $\vee$ *morphine* $\mathcal{X}_2$, establishing that $\mathcal{X}_1$ includes $\mathcal{X}_2$. Since $p_1 \neq p_2$, rules r_1 and r_2 conflict, and consequently, articles a_1 and a_2 also conflict.

Example 3. In Figure 1, Narcotics Control Act Article 60 mentions Article 2 of the same act.

A.2 Why ReX is powerful in LACD?

In this section, we explain the effectiveness of the ReX method in the LACD task in terms of the transitive structure of conflicts among legal articles. Consider a query article a_q, a conflicting article a_1 detected by the reranker, and another article a_2 known to conflict with a_1. Moreover, if the following three conditions hold, then *conflict*(a_q, a_2) is *guaranteed*:

1. a_1 contains exactly one rule r_1 (e.g., Criminal Act Article 205 in Figure 1).

$$r_1 = \textit{rule}(p_1, \mathcal{X}_1) \sqsubseteq a_1$$

2. There exist rules

$$r_q = \textit{rule}(p_q, \mathcal{X}_q) \sqsubseteq a_q$$
$$r_2 = \textit{rule}(p_2, \mathcal{X}_2) \sqsubseteq a_2$$

such that either $\mathcal{X}_2 \supset \mathcal{X}_1 \supset \mathcal{X}_q$ or $\mathcal{X}_q \supset \mathcal{X}_1 \supset \mathcal{X}_2$, inducing a conflict.

3. $p_q \neq p_2$.

Proof. Here, *rule*$(p_q, \mathcal{X}_q)$ and *rule*$(p_2, \mathcal{X}_2)$ *conflict*, as $\mathcal{X}_2 \subset \mathcal{X}_{\text{II}}$ or $\mathcal{X}_q \subset \mathcal{X}_{\in}$; and $p_q \neq p_2$ (from the condition 2 and 3). From the definition 2-2 in Section 2.1, *conflict*(a_q, a_2). ■

For example, if a_q conflicts with a_1 because it adjudicates a strict subset of cases, then any larger article a_2 whose scope includes that of a_1 will also conflict with a_q.

To validate this empirically on our dataset $\mathcal{S}$, we define and compute:

- P_1: the probability that a randomly chosen pair $(a_1, a_2) \in \mathcal{S}$ is conflict;
- P_2: the conditional probability that (a_2, a_3) is conflict given $(a_1, a_2) \in \mathcal{C}$;
- P_{TC}: the conditional probability that (a_1, a_3) is conflict given both $(a_1, a_2), (a_2, a_3) \in \mathcal{C}$.

We obtain

$$P_1 = 9.43\%, \quad P_2 = 21.2\%, \quad P_{\text{TC}} = 70.4\%.$$

This dramatic increase in P_{TC} stems from the transitive conflict relationships inherent in LACD, explaining why ReX is especially powerful in this domain.

A.3 Justification of Step 2-2 in ReX

In this section, we justify article selection policy in Step 2-2, which is determined as follows, where $\min = \text{minimum}(\{\textit{rerank}(a_q, a_i) | a_i \in \mathcal{A}_{\textit{topk}}\})$.

$$\mathcal{A}_{\textit{filter}} = \{a_i \in \mathcal{A}_{\textit{topk}} | \textit{rerank}(a_q, a_i) > \min / P_{TC}\}$$

In the following explaination, we simply denote *conflict*(a_q, a_i) as $c(a_q, a_i)$.

Justification. For the query article a_q and the retrieved article $a_1 \in \mathcal{A}_{\textit{topk}}$, our goal is to determine whether a_2 conflicts with a_q without using a

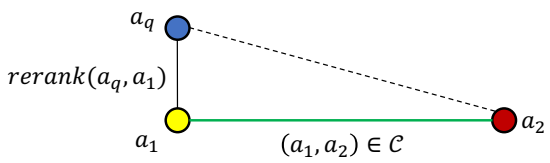

Figure 7: An example of article expansion in ReX.

reranker. As described in Section 2.2, the reranker returns a probability that represents the likelihood of a conflict between a_q and a_i. Ideally, we can interpret $P(c(a_q, a_1)) = rerank(a_q, a_1)$. By Bayes' theorem:

$$\begin{aligned}&P(c(a_q,a_2))\\&=P(c(a_q,a_1)\wedge c(a_1,a_2))\cdot P(c(a_q,a_2)|c(a_q,a_1)\wedge\\&\quad c(a_1,a_2))+P(\neg(c(a_q,a_1)\wedge c(a_1,a_2)))\cdot\\&\quad P(c(a_q,a_2)\mid\neg(c(a_q,a_1)\wedge c(a_1,a_2)))\end{aligned}$$

By ignoring the case where the triadic closure assumption does not hold, we can derive a conservative lower bound as follows:

$$\begin{aligned}&P(c(a_q,a_2))\geq\\&P(c(a_q,a_1)\wedge c(a_1,a_2))P(c(a_q,a_2)|c(a_q,a_1)\wedge c(a_1,a_2))\end{aligned}$$

Our main idea is to apply a *naïve Bayes approximation*, assuming independence between conflicts. Under this assumption, we can write:

$$P(c(a_q,a_1)\wedge c(a_1,a_2))=P(c(a_q,a_1))P(c(a_1,a_2))$$

Furthermore, we generalize $P(c(a_q, a_2) \mid c(a_q, a_1) \wedge c(a_1, a_2))$ to P_{TC}, as introduced in Section 3 and Appendix A.2. Therefore:

$$P(c(a_q, a_2)) \geq rerank(a_q, a_1) \cdot P_{TC}$$

In Step 2-2, our goal is to selectively expand $\mathcal{A}_{topk}$. To ensure that the expected probability for each selected a_2 is greater than the minimum probability in $\mathcal{A}_{topk}$ (i.e., $\min$), the following must hold:

$$P(c(a_q, a_2)) \geq rerank(a_q, a_1) \cdot P_{TC} > \min$$

Hance, we conclude:

$$rerank(a_q, a_1) > \min / P_{TC} \qquad \blacksquare$$

A.4 Data collection criteria

1. Article a_1 is in the *Criminal Act* and has a mention relationship with a_2, or vice versa.
2. Both a_1 and a_2 appear in one of the *acts about crimes*.
3. Either criteria 1 or 2 holds for a_1, and a_2 has a high similarity score with a_1 according to *enc-bi*.

The term *acts about crimes* means following acts. These are selected based on the Korean Bar Exam guidelines[3]:

- Criminal Act
- Act on Special Cases Concerning the Punishment of Sexual Crimes
- Act on the Aggravated Punishment of Specific Economic Crimes
- Act on the Aggravated Punishment of Specific Crimes
- Punishment of Violences Act
- Act on the Protection of Children and Youth Against Sex Offenses

A total of 1,081 pairs were collected based on the first criterion, while the remaining 1,172 and 1,921 pairs were gathered using the second and the third criterion, respectively.

A.5 Conflicts in the real world

A.5.1 Hierarchy of laws

In Korea, a legal article is included in **Acts** if and only if the article is enacted by national assembly of Korea. Otherwise, it is classified differently (e.g., enforcement degree, enforcement rule). There exists a hierarchy among Acts, enforcement decrees, and enforcement rules, with Acts being the most authoritative. In Korea, if two legal articles of differing hierarchy conflict, the lower article must be ignored. In this study, we exclusively focus on articles within Acts, and LMGraph contains 79,615 articles that meet this criterion.

A.5.2 Solving conflicts in Korea

In Korea, if articles a_1 and a_2 conflict with each other, and able to judge some case c, one of them is invalidated (i.e., ignored in the judgment). There are two principles to solve conflicts as follows[4]:

1. A new law overrides an old law (*lex posterior derogat priori*)

[3] Supplementary Acts for Bar Exam, Ministry of Justice, 2011.

[4] The supreme court of Korea, 88누6856, 1989. 9. 12.

2. A specific law overrides a general law (*lex specialis derogat leges generales*)

We explain each principle in Example A.1 and Example A.2, respectively.

Example A.1. Criminal Act 201 and Narcotics Act 60. As we explained in Section 1, Criminal Act 201 and Narcotics Act 60 conflict with each other, and thus a crime of using opium is judged by both articles. In terms of time, Criminal Act 201 is relatively old (enacted in 1953) than Narcotics Act 60 (enacted in 2000). Thus, according to principle (1), Narcotics Act 60 overrides Criminal Act 201 (i.e., Criminal Act 201 is ignored in this case).

Example A.2. Criminal Act 201 and Criminal Act 205. For the case c_1 in Example 1, Section 2.1, we can apply not only Criminal Act 205, but also Criminal Act 201 because bob smoked (the same as *used*) opium in his house. Therefore, Criminal Act 201 and 205 are conflict with each other and able to judge c_1. From the descriptions of each article, Criminal Act 205 judges more general cases than Criminal Act 201 (details are in Example 2, Section 2). Thus, according to principle (2), Criminal Act 201 overrides Criminal Act 201 (i.e., Criminal Act 201 is ignored in this case).

A.5.3 Quality evaluation question

We consulted legal experts to verify two aspects of the constructed LACD dataset: (1) whether pairs were correctly labeled concerning conflicts under Korean criminal law, and (2) whether pairs accurately met the defined criteria for Legal Article Conflicts. As a result, we confirmed that 94% of the dataset pairs were correctly constructed according to criterion (1), and 100% were correctly labeled according to criterion (2).

A.5.4 Articles without rules

Some statutes do not adjudicate real cases; instead, they merely define specific terms or state the purpose of the Act. Such statutes therefore do not contain rules. For example, in Figure 1, Article 2 of the Narcotics Act only defines terminology and thus does not contain any rules that adjudicate cases. In our dataset, 47 out of 350 articles of the Korean Criminal Act (13.4%) do not contain rules. Although Definition 2 in Section 2.1 allows for the possibility that an article without rules could conflict with another article, such conflicts are beyond the focus of this paper.

A.6 Details of training and testing

Notations: In Section 4.1 and Section 4.3, we define two sets of article pairs, $\mathcal{C}$ and $\mathcal{S}$, which are derived from the test and validation sets, respectively. These sets are defined as follows:

- $\mathcal{S}$: Identical to the training set. It includes both conflicting and non-conflicting article pairs.
- $\mathcal{C}$: Consists of **only** conflicting pairs drawn from the training and test sets. For $(a_1, a_2) \in \mathcal{C}$, we augment $\mathcal{C}$ by adding (a_2, a_1) in $\mathcal{C}$ (commutative law).

In the LACD dataset, the sizes of the sets are $|\mathcal{S}| = 226 + 2{,}278 = 2{,}504$ and $|\mathcal{C}| = 690$ (see Table 2).

Hyperparameters: Table 6 shows the settings of our experiments.

Setting	Value
General settings	
Optimizer	Adam (Kingma and Ba, 2015)
Warmup steps	500
Weight decay	0
Batch size per device	16
SEED	0, 1, 2
Training *enc-cross*	
learning rate	$5 \cdot 10^{-5}$
epochs	10
ReX	
P_{TC}	0.704
Others	
Context length	512 (Klue/RoBERTa) 2048 (KoBigBird)
prob in Step 3	one layer *FFNN* and sigmoid function

Table 6: Summary of experimental settings. Here, *FFNN* means Feed Forward Neural Networks.

Model size and computational resources: The two models in our experiments, KoBigBird (*enc-bi*) and Klue/RoBERTa (*enc-cross*), contain 114 million and 111 million parameters, respectively. All experiments are conducted on a single machine equipped with eight NVIDIA TITAN RTX GPUs. We train four different rerankers, each with three independent runs. The total computational cost amounts to 16 GPU hours on an NVIDIA TITAN RTX.

A.7 LGNN reranker for other Korean legal NLP benchmarks

As discussed in Section 6, there is currently no publicly available benchmark for article retrieval

in the Korean legal NLP domain. Thus, to further evaluate the generalizability and effectiveness of the proposed LGNN reranker beyond the LACD task, we apply it to a different Korean legal NLP benchmark. Specifically, we utilize the `statute_classification_plus` dataset from the LBox-Open benchmark (Hwang et al., 2022), which is originally formulated as a multi-label classification problem. We converted it into a binary classification setting to align it with our reranking step.

Table 7 shows the performance comparison between naïve and LGNN reranker at the LBox-Open dataset. Our LGNN reranker achieved F1 score improvement of 2.5%p, and The results show the effectiveness of the LGNN reranker in other Korean legal NLP tasks.

Reranker method	F1	Acc.	ROC AUC
Re2 reranker	78.8	91.6	94.2
LGNN reranker **(Ours)**	**81.3**	**93.4**	**95.3**

Table 7: Performance (%) of LGNN reranker on the `statute_classification_plus` task in the LBox-Open benchmark. We use KLUE/RoBERTa-base as both *enc-cross* and *enc-bi*. Since the original dataset contains only positive (true) pairs, we generate negative (false) pairs by using high BM25-scored articles. We utilize 10% of training pairs for training reranker.

A.8 Evaluation metrics

We evaluate retrieval performance using three macro-averaged metrics at various cut-off levels n: Recall@n, nDCG@n, and Retrieval F1@n. Let $\mathcal{Q}$ be the set of all queries (size $N_Q = |\mathcal{Q}|$), and for each query $a_q \in \mathcal{Q}$, let:

- Rel_{a_q} be the set of true (relevant) articles,
- $\mathsf{Ret}_{a_q}^{(n)}$ be the set of top-n retrieved articles.

Macro-Recall@n: For each query a_q, the Recall@n is

$$\text{Recall@}k(a_q) \;=\; \frac{\left|\mathsf{Ret}_{a_q}^{(k)} \cap \mathsf{Rel}_{a_q}\right|}{\left|\mathsf{Rel}_{a_q}\right|}.$$

Then, the macro-Recall@n is:

$$\text{Macro-Recall@}n \;=\; \frac{1}{N_Q}\sum_{a_q\in\mathcal{Q}} \text{Recall@}k(a_q)$$

Macro-nDCG@n: Define the Discounted Cumulative Gain (DCG) for query a_q at rank n as

$$\text{DCG@}n(a_q) \;=\; \sum_{i=1}^{n} \frac{2^{\text{rel}_{a_q,i}} - 1}{\log_2(i+1)}$$

where $\text{rel}_{q,i} = 1$ if the i-th retrieved item is relevant, and 0 otherwise. Let $\text{IDCG@}n(a_q)$ denote the maximum possible DCG@n under an ideal ranking. Then

$$\text{nDCG@}n(a_q) \;=\; \frac{\text{DCG@}n(a_q)}{\text{IDCG@}n(a_q)}$$

and the Macro-nDCG@n is:

$$\text{Macro-nDCG@}n \;=\; \frac{1}{N_q}\sum_{a_q\in\mathcal{Q}} \text{nDCG@}n(a_q)$$

Macro Retrieval F1@n: For each query a_q, define

$$P@n(a_q) = \frac{\left|\mathsf{Ret}_{a_q}^{(n)} \cap \mathsf{Rel}_{a_q}\right|}{n}.$$

Then the per-query F1@k is

$$F1@n(a_q) \;=\; \frac{2\,P@n(a_q)\,Recall@n(a_q)}{P@n(a_q) + Recall@n(a_q)}$$

and the macro-retrieval F1@n is

$$\text{Macro-F1@}n \;=\; \frac{1}{N_Q}\sum_{q\in\mathcal{Q}} F1@n(a_q)$$

A.9 Significance test for the main results

To assess whether the observed improvements of GReX over the baseline Re2 are statistically significant, we conduct paired Student's t-tests for each metric, following the procedure described in Urbano et al. (2019). The null hypothesis H_0 assumes no performance difference between GReX and Re2, while the alternative hypothesis H_1 assumes a difference exists. Formally, we define:

$$H_0 : \mu_\Delta = 0 \quad \text{(no difference)}$$
$$H_1 : \mu_\Delta \neq 0 \quad \text{(significant difference)}$$

where $\Delta_i = X_i^{\text{GReX}} - X_i^{\text{Re2}}$ denotes the paired per-query difference between the two systems.

For each metric, we use the `scipy.stats.ttest_rel` function to perform a paired t-test. Table 8 shows p values for each metric. From the results, for all metrics, the two-tailed p-values are below the significance threshold of 0.05. Therefore, we reject the null hypothesis and conclude that GReX significantly outperforms Re2 across all evaluation metrics.

Metric	GReX	Re2	p
nDCG@5	25.27 ± 0.95	16.38 ± 1.48	0.0332 *
nDCG@10	30.91 ± 1.51	20.38 ± 1.97	0.0376 *
nDCG@50	37.79 ± 1.00	26.09 ± 0.98	0.0138 *
Recall@5	34.88 ± 1.35	24.00 ± 1.65	0.0212 *
Recall@10	50.34 ± 2.83	35.15 ± 3.64	0.0224 *
Recall@50	76.65 ± 1.05	57.71 ± 1.34	0.0078 *
Retrieval F1@5	16.82 ± 0.57	12.01 ± 0.81	0.0069 *
Retrieval F1@10	13.99 ± 0.90	9.91 ± 0.83	0.0295 *
Retrieval F1@50	5.13 ± 0.07	3.67 ± 0.09	0.0059 *

Table 8: Paired t-test results comparing GReX and Re2. Asterisks denote $p < 0.05$.