# Granularity-aware legal question answering: a case study of indonesian government regulations

Douglas Raevan Faisal [a,b,1,*], Fariz Darari [a,b,2], Reynard Adha Ryanda [a,3]

[a] Faculty of Computer Science, Universitas Indonesia, Depok 16424, Indonesia
[b] Tokopedia-UI AI Center of Excellence, Universitas Indonesia, Depok 16424, Indonesia
[1] douglas.raevan11@ui.ac.id; [2] fariz@cs.ui.ac.id; [3] reynard.adha@gmail.com
* corresponding author

ARTICLE INFO

**Article history**
Received May 11, 2023
Revised May 16, 2024
Accepted May 17, 2024
Available online August 31, 2024

**Keywords**
Granularity-aware
Question answering
Retrieval
Regulation
BERT

ABSTRACT

Question answering (QA) technologies are crucial for building conversational AI. Current research related to QA for the legal domain lacks focus on the organized structure of laws, which are hierarchically segmented into components at varying levels of detail. To address this gap, we propose a new task of granularity-aware legal QA, which accounts for the underlying granularity levels of law components. Our approach encompasses task formulation, dataset creation, and model development. Under the Indonesian jurisdiction, we consider four law component granularity levels: chapters (bab), articles (pasal), sections (ayat), and letters (huruf). We include 15 government regulations (Peraturan Pemerintah) of Indonesia related to labor affairs and build a legal QA dataset with granularity information. We then design a solution for such a task—the first IR system to account for legal component granularity. We implement a customized retriever-reranker pipeline in which the retriever accepts law components of multiple granularities and the reranker is trained for granularity-aware ranking. We leverage BM25 and BERT models as retriever and reranker, respectively, yielding an end-to-end exact match accuracy of 35.68%, which offers a significant improvement (20%) over a strong baseline. The use of reranker also improves the granularity accuracy from 44.86% to 63.24%. In practical context, such a solution can help provide more precise answers, not only from legal chatbots, but also other conversational AI that deals with hierarchically-structured documents.



## 1. Introduction

Conversational Artificial Intelligence (AI) is gaining traction among the public and enterprises. It enables better human-machine interaction through conversation. Such a technology can be realized by integrating multiple aspects of Natural Language Processing (NLP) for Question Answering (QA) and Information Retrieval (IR).

One of the primary goals of conversational AI is to provide natural interaction. As such, it is essential for conversational AI platforms to mimic how humans give answers when faced with questions from users while retaining the functionalities of providing the correct information to the questions. The computational model behind a question-answering process has been proposed by [1]. More recent and notable endeavors in QA include datasets and benchmarks such as SQuAD (Stanford Question

Answering Dataset) [2], MS MARCO (Microsoft Machine Reading Comprehension) [3] and HotpotQA [4], along with QA models such as BERT [5] and dense passage retrieval [6].

Within the legal context, the use of conversational AI can enhance user experience and increase the cost-effectiveness in user inquiries around laws under certain jurisdictions. The implementation of legal chatbots has been explored by Lynx/LynxSP [7], [8], LAW-U [9], Lex2KG-based virtual assistant [10], and an Indian-based chatbot developed by [11]. With the help of electronic discovery over legal data [12] and legal question-answering (QA) technologies [13], these legal chatbots can answer questions about laws and highlight the use of NLP, IR, and knowledge graph (KG) in providing their services.

Dealing with legal documents requires consideration of the underlying structure on which they are written (e.g., chapters, articles, and sections), unlike regular text documents. When discussing legal affairs, it is also necessary to include accurate and complete law references as evidence. As a result, developing a legal QA system can benefit from the hierarchical structure.

Taking the above into consideration, a challenge arises as to which law component granularity has the best contextual coverage to answer a given question. Table 1 demonstrates a couple of interactions related to the government regulation on "Foreign Workers Utilization" (PP 34/2021), highlighting the importance of granularity in legal QA.

**Table 1.** Examples of correct answers but with incorrect granularity

| English | Indonesian |
|---|---|
| ***Interaction 1*** | |
| **Q:** Can employers employ foreign workers to take care of personnel affairs?<br>**A:** [Article 11 section (1) PP 34/2021]<br>Employer of TKA* is prohibited from employing TKA* in positions in charge of personnel affairs. | **Q:** *Apakah boleh pemberi kerja TKA* mempekerjakan TKA* untuk mengurusi personalia?*<br>**A:** [*Pasal* 11 *ayat* (1) PP 34/2021]<br>*Pemberi Kerja TKA* dilarang mempekerjakan TKA* pada jabatan yang mengurusi personalia.* |
| **Note:** The answer given here is incomplete, as it should also include the next section, which describes the job positions mentioned. Since Article 11 only consists of 2 sections, it would be better answered with the article altogether. | |
| ***Interaction 2*** | |
| **Q:** What is the deadline for submitting an application for an extension of RPTKA** legalization?<br>**A:** [Article 21 PP 34/2021]<br>Application for extension of Legalization of the RPTKA is submitted online by the Employer of TKA to the Minister or appointed official.<br>Application for extension of Legalization of the RPTKA** as referred to in section (1) is submitted not later than 30 (thirty) work days before the expiration date.<br>...<br>The Minister or appointed official as referred to in section (5) transmits data of TKA to be employed online to the minister administering government affairs in the field of law and human rights or the recommended appointed official to obtain a stay permit to work. | **Q:** *Kapan paling lambat permohonan perpanjangan pengesahan* RPTKA** *perlu diajukan?*<br>**A:** [*Pasal* 21 PP 34/2021]<br>*Permohonan perpanjangan Pengesahan* RPTKA** *diajukan oleh Pemberi Kerja* TKA* *secara daring kepada Menteri atau pejabat yang ditunjuk.*<br>*Permohonan perpanjangan Pengesahan* RPTKA** *sebagaimana dimaksud pada ayat* (1) *diajukan paling lambat* 30 (*tiga puluh*) *hari kerja sebelum jangka waktu berakhir.*<br>...<br>*Menteri atau pejabat yang ditunjuk sebagaimana dimaksud pada ayat (5) menyampaikan data* TKA* *yang akan dipekerjakan secara daring kepada menteri yang menyelenggarakan urusan pemerintahan di bidang hukum dan hak asasi manusia atau pejabat yang ditunjuk sebagai rekomendasi untuk mendapatkan izin tinggal dalam rangka bekerja.* |

[a.] Note: The question can be answered more precisely by just section (2) of Article 21.

[b.] TKA stands for tenaga kerja asing which translates to foreign workers.

[c.] RPTKA stands for rencana penggunaan tenaga kerja asing which translates to foreign workers utilization plan.

The first interaction suggests that Article 11 section (1) may seem like the correct answer for the given question. However, it turns out that Article 11 section (2) is also relevant and cannot be missed out. Since the article only consists of two sections, the question is better answered with Article 11 altogether. On the other hand, the second interaction illustrates the other extreme: an answer that is too broad in scope is given when a precise answer is actually needed. In this case, Article 21 section (2) already suffices to answer the given question.

Before we highlight our contributions, we first cover related work prior to this research. We first touch on the definitions of QA and IR. Then, we explain several endeavors related to answer granularity classification. Finally, we describe other studies in legal information retrieval and extraction that are relevant for this research.

### 1.1. Question Answering and Information Retrieval

Computational QA goes back as far as the early 1960s ever since computers were invented [14]. Lehnert [1] studied how a computational model can be designed to mimic the process of human answering. QA systems can act as a virtual assistant and can be integrated with a search engine or database [14]. In general, given some sense of information or knowledge, a QA system can answer questions based on the supplied information.

Information Retrieval (IR) is defined as the task of "finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)" [15]. A pipeline of IR and QA results in a new task of open domain QA which employs IR methods to retrieve the relevant set of documents and then uses a reading comprehension model to extract the text spans of those relevant documents that best answer the given question. This paradigm is largely found in most QA systems based on the retriever-reader architecture [6], [16], [17].

### 1.2. Answer Granularity Classification

An answer to a question should contain just enough information to complete the missing knowledge based on the question and be linguistically comprehensible. The topic of answer granularity has become the main discussion of the AQUAINT Project and its applications have been investigated in [18] and [19]. Allan et al. [19] have discussed the different levels of textual granularity for conversational answer retrieval. AbdulJaleel et al. [18] leveraged this feature as metadata to support the retrieval and question answering at Text Retrieval Conference (TREC) 2003. They incorporated multiple levels of textual granularity, which are sentence, passage, and document.

Lehnert [1] introduced question classification as the first pass of the QA system. The proposed question types are: (1) why questions, (2) how questions, (3) yes or no questions, (4) occurrence questions, and (5) component questions. Bu et al. [20] classified questions based on their functions into six classes: (1) fact, (2) list, (3) reason, (4) solution, (5) definition, and (6) navigation. A non-factoid QA taxonomy is proposed in [21] consisting of six categories: (1) instruction, (2) reason, (3) evidence-based, (4) comparison, (5) experience, and (6) debate. ResPubliQA [22], a multilingual QA evaluation task over European legislation, grouped their question collections over legal cases into six classes: (1) factoid, (2) definition, (3) reason/purpose, (4) procedure, (5) opinion, and (6) other.

Based on the previous types of questions from different taxonomies, we map question types to the answer granularity as shown in Table 2. Starting from the smallest granularity, word(s)/phrase(s) answers are resulted from yes or no questions [1], component [1], fact/factoid [20], and list [20]. These question

types do not require a complete sentence structure of subject-predicate-object and can be answered with a single yes/no word, a single entity, or multiple entities.

**Table 2.** Mapping Question Types to Answer Granularity

| Question Types | Examples | Answer Granularity | | |
|---|---|---|---|---|
| | | *Word(s)/ Phrase(s)* | *Sentence* | *Passage* |
| Why [1]<br>Reason/Purpose [20]–[22] | *Why did John go to New York?; Why is the earth round?* | | ✓ | ✓ |
| How [1]<br>Solution [20]<br>Instruction [21]<br>Procedure [22] | *How to cook pasta?* | | | ✓ |
| Yes or No [1] | *Did John go to New York?; Are the rooms clean?* | ✓ | | |
| Occurrence [1] | *What did John do in New York?* | | ✓ | ✓ |
| Component [1]<br>Fact/Factoid [20], [22]<br>List [20] | *Where did John go this weekend?; What are the ingredients to make a lasagna?* | ✓ | | |
| Definition [20], [22]<br>Evidence-Based [21] | *What is a peninsula?; What is the meaning of 'bootstrapping'?* | | ✓ | ✓ |
| Experience [21] | *Would you recommend visiting Tokyo?* | | ✓ | ✓ |
| Debate [21]<br>Opinion [22] | *Is Pluto considered as a planet?* | | | ✓ |

[d] https://ciir.cs.umass.edu/aquaint

Sentence answers are provided from why questions [1], reason questions [20], [21], [22], definition [20], [22], evidence-based questions [21], and experience questions [21]. These types of questions require at least a complete sentence structure to answer and not necessarily a text passage of multiple sentences.

Lastly, longer answers (passage) are provided by answering why/reason questions [1], [20], [21], how/solution/instruction questions [1], [20], [21], [22], occurrence [1], definition [20], [21], [22], experience [21], debate [21], and opinion [22]. These questions need comprehensive answers which may include multiple sentences.

An issue with answer granularity classification is that different questions with the same question type may be answered with different granularities, especially in informal conversational settings where sentence structure is often disregarded. We recognize this issue and acknowledge that the classification as shown in Table 2 may vary depending on the occasion. In this paper, we only focus on the granularities of law components since the structures are already pre-defined in the law documents.

### 1.3. Law Representation and Retrieval

The Competition on Legal Information Extraction/Entailment (COLIEE) is an annual competition that focuses on the information retrieval and entailment aspects of legal texts [23]. One of the tasks—statute law retrieval—requires the model to retrieve a correct subset of Japanese Civil Code Articles (consisting of 768 articles) given a Japanese legal bar exam question. The participants explored varying approaches, ranging from simpler techniques (e.g., BM25 and TF-IDF) up to deep-learning-based techniques (e.g., BERT [5], Sentence-BERT [24], LEGAL-BERT [25], and Ensemble BERT [26]).

Another similar competition—ResPubliQA consists of QA tasks related to legal domain, focusing on retrieving paragraphs from a collection of the European legislation [22].

Another recent initiative in legal information retrieval is legal community question answering. In this field, information retrieval leverages publicly available sources. For instance, FALQU [27] was created based on the questions and answers of legal forum users. The dataset is then benchmarked against several baseline systems for retrieval, such as BM25, TF-IDF, and Sentence-BERT [24]. On the other hand, LegalQA [28] merges the questions of everyday users with answers from identified lawyers. Legal answer retrieval on the dataset was explored using BM25 as retrievers and cross-encoders as rerankers.

Knowledge Graph (KG) is described as a graph representation of knowledge in the real world with nodes representing entities of interest and edges representing the relations between entities [29]. KGs have gained traction in representing legal knowledge [6], [30]. One of the prominent projects in delivering legal KG services is Lynx [7] and Lynx Services Platform [8] as part of the European Law Identifier (ELI) initiative by EUR-Lex . Linked data representations for legal matters are also leveraged in several studies, such as event extraction in labor law [31] and similar case prediction [32]. Another endeavor, Lex2KG [33], provides a legal ontology that fits the Indonesian jurisdictions. This was then explored further by implementing a legal knowledge virtual assistant (VA) [10] to add the capabilities of answering user questions related to the labor law (Act 13/2003 concerning Manpower and its amendment in Act 11/2020 concerning Job Creation) in Indonesia.

One notable issue in the previous work [10] was related to the law components' granularities. In the most basic form, passages in a single law are coded into an article (pasal). Some articles are broken down into several sections (ayat) and even further into letter points (huruf). Conversely, multiple articles can also be grouped into a single chapter (bab). This issue presents an obstacle in legal question answering (QA) using law components—given a question, at what granularity level is the law component that best answers the user questions. It is also worth noting that existing retrieval evaluation related to legislation does not consider multiple granularities: ResPubliQA retrieves paragraphs [22], whereas COLIEE retrieves a subset of articles [21].

### 1.4. Contributions

Based on the issues as exemplified above, we propose our work on granularity-aware legal question answering. In particular, our contributions in this paper are as follows:

- We introduce the task of granularity-aware legal question answering.
- We create a dataset of granularity-aware legal QA based on fifteen of the recent Government Regulations of the Republic of Indonesia (Peraturan Pemerintah Republik Indonesia) regarding labor affairs consisting of 739 question-answer pairs.
- We develop the first solution based on a retriever-reranker pipeline optimized for granularity-aware legal QA.
- We perform experiments using a variety of retriever and document ranking models based on BM25 and BERT.

The remainder of this paper is organized into the following sections: Section 2 formally defines the task of granularity-aware legal QA and describes the research methodology followed by explaining the development process; Section 3 presents the research findings in implementing the granularity-aware legal QA and discusses possible future directions; and Section 4 concludes the research.

## 2. Method

In this section, we first formulate the granularity-aware legal QA task, followed by describing the methodology of our research. Next, we list our dataset of regulation documents and question-answer pairs. Afterwards, we detail our retriever-reranker models and their evaluation settings.

### 2.1. Task Description and Formulation

We propose the task of "Granularity-Aware Legal Question Answering". This task introduces the granularity aspect into law retrieval tasks as described in [22] and [23]. Since this work is based on giving a better conversational AI experience for the users, we intend to make the task's output to be one law component for each given question, similar to [22]. The answer is selected from several law components as candidates based on a certain criterion that will be elaborated below.

Given a question q and a set of law components $D = \{l_1, l_2, \dots, l_n\}$ of size $N$, such that $l_i$=($p_i, g_i$ ) is a law component where $p_i$ is the text passage of the law component and $g_i$ is the granularity level of that law component, the task of the granularity-aware legal question answering is defined as.

$$f(q) = argmax\ combine\ (text_{sim(q,p_i)}, granularity_{match(q,p_i,g_i)}) \quad (1)$$

where we intend to find the law component $l^* = (p^*, g^*)$ in the set $D$ that maximizes the combined value of two different objectives: text similarity and granularity match. The combine function described above covers the general description of several aggregate operators, such as addition, weighted addition, multiplication, weighted multiplication, and conditional operators. The first value to be combined is text similarity (text_sim), which refers to the common measure found in IR ranking methods. It assesses how textually similar the query and the law component are. The novelty of this research is the granularity match value (granularity_match) in addition to the text similarity. The value signifies whether the query can be fulfilled with the given law component's granularity and text passage.

For this task, four levels of granularity are included, as most laws and regulations have at least these levels: (1) chapter (*bab*), (2) article (pasal), (3) section (*ayat*), and (4) letter (*huruf*). Relating to the discussion on answer granularity in [18] and [19], law component granularities can be mapped to answer granularities in the following ways: chapters to passages; articles to either passages or sentences; sections to sentences; and letters to short answers.

In this paper, we will often use the term granularity path. It refers to determining whether two components, i.e., component A and component B, are related to each other based on whether one component is part of another component but with different granularity levels. For example, Article 3 section (2) is part of Article 3, and thus, Article 3 and Article 3 section (2) are in the same granularity path. On the other hand, Article 3 is not part of Article 4, hence, they are not in the same granularity path

### 2.2. Methodology

This research is conducted in several steps. Following the research question posed, we first formulate the task of granularity-aware legal QA (as described in the previous subsection). Such a task was formulated in order to formalize the objective in which the developed QA model should follow.

In this study, we develop two novel datasets for the task: multi-granularity regulation documents and question-answer dataset. The former is collected by selecting several government regulations under a single topic (that is, labor affairs) and converting them into structured data using Lex2KG [33].

Meanwhile, the question-answer dataset is constructed by performing manual question generation and annotations for the included government regulations.

Finally, we design and implement several models optimized for granularity-aware legal QA by leveraging IR and NLP techniques. We rely on the retriever-reranker pipeline to decouple the different objectives of performing document retrieval and granularity-aware reranking. We also experiment with different combinations of NLP preprocessing and feature engineering, such as query expansion, stopwords removal, and stemming. We evaluate and analyze these models to get insights on how well the models can handle this novel task. Our research focuses on the experimental study of different methods and techniques to build a granularity-aware legal QA model.

### 2.3. Regulation Documents

For this work, we include fifteen different labor-related government regulations from the year 2015 to 2022. These regulations represent the legal jurisdiction on labor affairs in Indonesia. We use an online database for laws and regulations provided by BPK RI (Audit Board of the Republic of Indonesia). We choose the labor/manpower (ketenagakerjaan) as the main topic, and then select regulations that are released within the last 10 years and are still in force. The selected regulations are listed in Table 3. Some of these regulations are also direct implementations of the Omnibus Law on Job Creation [34].

**Table 3.** List of Included Labor Regulations

| Regulation | Title (Indonesian) | Title (English) |
|---|---|---|
| PP 4/2015 | *Pelaksanaan Pengawasan Terhadap Penyelenggaraan Penempatan Dan Perlindungan Tenaga Kerja Indonesia Di Luar Negeri* | Implementation of Supervision of the Implementation of Placement and Protection of Indonesian Migrant Workers Overseas* |
| PP 44/2015 | *Penyelenggaraan Program Jaminan Kecelakaan Kerja Dan Jaminan Kematian* | Work Accident and Casualty Security Program Implementation |
| PP 45/2015 | *Penyelenggaraan Program Jaminan Pensiun* | Pension Security Program Implementation |
| PP 46/2015 | *Penyelenggaraan Program Jaminan Hari Tua* | Old Age Security Program Implementation |
| PP 88/2019 | *Kesehatan Kerja* | Occupational Health* |
| PP 10/2020 | *Tata Cara Penempatan Pekerja Migran Indonesia oleh Badan Pelindungan Pekerja Migran Indonesia* | Procedures for Placement of Indonesian Migrant Workers by Indonesia Migrant Workers Protection Board |
| PP 49/2020 | *Penyesuaian Iuran Program Jaminan Sosial Ketenagakerjaan Selama Bencana Nonalam Penyebaran Corona Virus Disease 2019 (COVID-19)* | Adjustment of Employment Social Security Program Contributions During Non-Natural Disasters Spread of Corona Virus Disease 2019 (COVID-19)* |
| PP 60/2020 | *Unit Layanan Disabilitas Bidang Ketenagakerjaan* | Disability Services Unit in the Employment Sector* |
| PP 34/2021 | *Penggunaan Tenaga Kerja Asing* | Foreign Workers Utilization |
| PP 35/2021 | *Perjanjian Kerja Waktu Tertentu, Alih Daya, Waktu Kerja dan Waktu Istirahat, dan Pemutusan Hubungan Kerja* | Fixed Term Employment Contract, Outsourcing, Working Hour and Rest Period, and Termination Of Employment |
| PP 36/2021 | *Pengupahan* | Wages |
| PP 37/2021 | *Penyelenggaraan Program Jaminan Kehilangan Pekerjaan* | Administration of Job Loss Security Program |
| PP 59/2021 | *Pelaksanaan Pelindungan Pekerja Migran Indonesia* | Implementation of Protection for Indonesian Migrant Workers* |
| PP 94/2021 | *Disiplin Pegawai Negeri Sipil* | Civil Service Discipline* |
| PP 22/2022 | *Penempatan dan Pelindungan Awak Kapal Niaga Migran dan Awak Kapal Perikanan Migran* | Placement and Protection of Migrant Commercial Ship Crews and Migrant Fishing Ship Crews* |

[e.] There is no official English translation for these regulations. The titles are translated using Google Translate.

From these regulations, we manage to extract law components from the PDF documents using Lex2KG [33]. The extraction process comprises several steps: (1) optical character recognition (OCR), (2) scanning, (3) cleaning, (4) parsing, and (5) constructing. The extracted data is in the form of linked data, i.e., Resource Description Framework (RDF) triples [35]. The linked data represents the legal content and metadata of the regulations in a more machine-readable format.

To give a better view of the aspects and topics discussed in these regulations, we map the text passages of these law components into a word cloud as displayed in Fig. 1. Several dominant keywords are "peserta" (members, mainly refers to the members of insurance programs) "pekerja/buruh" (workers/labors), "BPJS Ketenagakerjaan" (National Social Security Agency for Employment), "pekerja migran Indonesia" (Indonesian migrant workers), and "pemberi kerja" (employers).


**Fig. 1.** Key Terms in Indonesian Labor Regulations Visualized in a Word Cloud

In total, there are 3,318 components, comprising 98 chapters, 630 articles, 1,258 sections, and 1,332 letters. It is worth noting that to fulfill the requirement of multi-granularity retrieval, the text passage of each component type may overlap with each other, since accounting for multiple granularities implies that, for instance, sections (as part of articles) are indexed into the same document collection as articles. This means that some components may have "part of" relationships to each other and, content-wise, are not necessarily disjoint.

### 2.4. Question-Answer Dataset

We construct a dataset for the granularity-aware legal QA task—a novel contribution for its kind consisting of 739 question-answer pairs along with the expected law component granularity of the answers. In total, five annotators with computer science background, who have been briefed for understanding legal documents, contributed to develop the question-answer pairs. For each regulation document, the annotators first read the whole document, and then create questions based on the content of the document. Out of those 739 question-answer pairs, 87 are answered with the granularity of chapters, 182 with articles, 283 with sections, and 187 with letters.

Over the dataset, we perform an exploratory data analysis (EDA) on the length of the questions (based on the number of words) for each expected answer granularity type. Based on Fig. 2, we find that questions answered with chapter tend to be shorter than the rest. We also observe that the interquartile range (IQR) is increasing with the more fine-grained granularity type, except for article vs. section. It indicates that the questions answered with smaller components are more varying in length. However, the bigger takeaway from this is that the task of classifying granularities is non-trivial as shown by the similar characteristics of the questions.

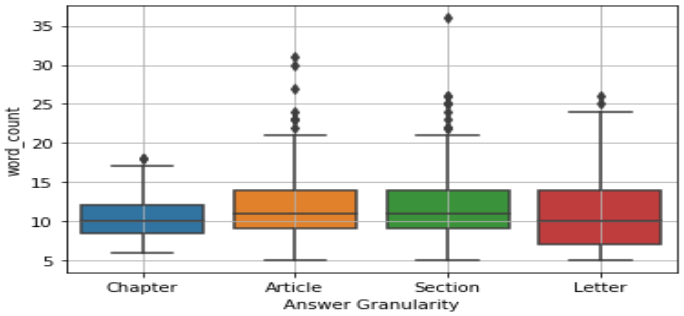


**Fig. 2.** Boxplot for Question Word Count by Answer Granularity Type

## 2.5. Modeling

We leverage a retriever-reranker pipeline to tackle the granularity-aware legal QA task as shown in Fig. 3. The retriever acts as a general-purpose ranker for the indexed law components, while the reranker performs granularity-wise ranking. For a given query, its answer is expected to be a single law component with the correct granularity that is the most relevant to the query. For example, as shown in Fig. 3, the input question is first reformulated as a search query (will be elaborated in a later part). The formulated query is then fed into the retriever to get the k most relevant documents, which, in this case, are Article 3 and its related components (assuming that: Article 3 is part of Chapter 2, Article 3 section (1) is part of Article 3, and Article 3 section (1) letter a is part of Article 3; and that the results are not strictly limited to components that are in the same granularity path as the expected answer). Afterwards, the reranker performs granularity-aware ranking, putting Article 3 section (1) as the most relevant answer.

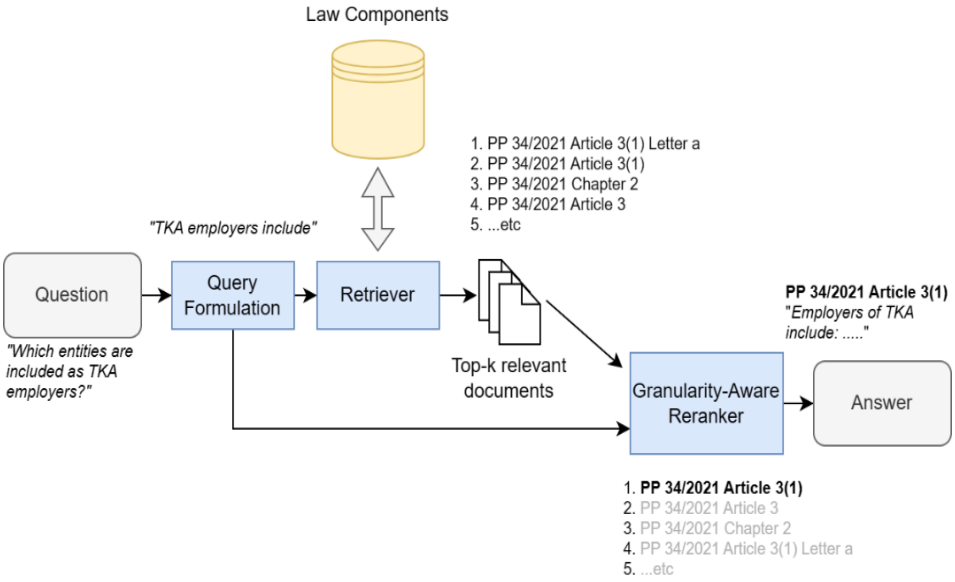


**Fig. 3.** Retriever-Reranker Pipeline for Granularity-Aware Legal QA

### 2.5.1. Retriever

For the retriever, we use the BM25 [36] implementation that comes with Elasticsearch, an open search and analytics engine. We also experiment with a more recent implementation of retriever, that is, Siamese-BERT [24], a siamese neural network architecture [37] with BERT as its subnetworks.

We apply two different indexing methods for BM25: articles-only and all-granularities. The articles-only indexing method will be used as the baseline to reflect the behavior of most retrieval systems where

the granularities are not regarded, whereas the all-granularities indexing method involves multiple granularities to be indexed. The latter method results in multiple documents with overlapping text passages, that is, the textual content of law components may contain that of lower-hierarchy components.

To improve the retrieval results, we employ several preprocessing techniques and feature engineering:

- Chapter title query expansion. Regulation documents are rich with metadata, and for this case, we utilize chapter titles to expand the query criteria.
- Question-to-statement conversion. We use question-to-statement conversion using pattern matching to reform the queries (most are written in interrogative form) into declarative sentences.
- Top-k parameter tuning. We try different k configurations for the top-k retrieval results prior to the reranking process.
- Stopwords removal and stemming. We experiment with simple text preprocessing techniques [38].

### 2.5.2. Granularity-Aware Reranker

Retrieval systems such as BM25 are not considered as a supervised ranking model as opposed to deep-learning-based models. Thus, BM25 could not be trained to perform granularity-aware ranking. We therefore leverage deep-learning-based models to rerank the retrieval results. The goal is to give higher ranks for the retrieved law component with the correct granularity levels. Previous research, such as [28], already implemented a retriever-reranker pipeline using BM25 and cross-encoders. However, the use of a reranker trained to understand the concept of different granularities is a novel approach in legal QA.

We use two different reranker architectures: Siamese-BERT [24] and Cross-Encoder BERT [5]. Siamese-BERT consists of two BERT models as encoders for each query and candidate document in which the outputs are measured based on similarity metrics (e.g., cosine similarity). As for the Cross-Encoder BERT, we refer to the use of BERT for retrieval by concatenating the query and the candidate document into one text sequence as the input. The model's architecture supports this method of input to accommodate the multi-genre natural language inference (MNLI) task [39] where a pair of hypotheses are fed into the model by concatenating them and classify whether one entails or contradicts another. To contrast, Siamese-BERT uses two separate BERT as encoders for each query and candidate document, while Cross-Encoder BERT relies on using BERT as a monolithic architecture to rank text passages.

For both architecures, we utilize two IndoBERT pretrained weights by Koto et al. [40] (indolem/indobert-base-uncased) and Wilie et al. [41] (indobenchmark/indobert-base-p2). Both pretrained models have been trained on a variety of Indonesian text corpora, such as Wikipedia articles, news articles, hotel reviews and social media posts. In this research, we try and compare the performance of each pretrained model to retrieve and rerank.

To train both architectures (Siamese and Cross-Encoder), we transform the training data into query-candidate pairs that are called examples. These examples are split into two sets: positive examples and negative examples. A positive example signifies when the candidate is relevant to answer the query, whereas a negative example represents a case where the candidate should not be included in the retrieval. Intuitively, the learning algorithm dictates the reranker model to better distinguish the positive examples from the negative examples.

In this case, we define a positive example as the expected answer for each query. We then experiment with three different training methods that are distinguished by the set of negative examples used during the training process:

- Relevance. For each training query, we select a few irrelevant law components at random that are not in the same granularity path as the expected answer (positive example). These will be transformed into a set of negative examples.
- Granularity. Instead of using entirely irrelevant candidates, the set of negative examples in this method consists of law components that are in the same granularity path as the expected answer (positive example).
- Relevance+Granularity. In this method, we combine both previous sets of negative examples.

### 2.6. Evaluation

The correctness of the retrieval results is evaluated by comparing the predicted law reference (e.g., "Article 8 section (2) PP 34/2021") to the predicted law reference to the expected law component's reference. Each question from the constructed dataset is used as input to the model to retrieve the predicted answer which is then compared with the expected answer. The final score would be the number of correct answers divided by the number of questions.

There are two different criteria for determining the correct answer. The first one is exact match (EM), which is to compare whether the predicted answer has the exact reference as the expected answer, like the one used in ResPubliQA [22]. On the other hand, the second criterion, article match (AM), is by only comparing the article number reference for each expected and predicted answer. It compares whether the expected and predicted answer has the same article in their respective granularity path. The latter criterion is less punishing than the first one in that the error of retrieving the correct article number but at the wrong level of granularity would be disregarded.

Table 4 demonstrates some example cases when using the two different criteria. Notice that when the exact match is true, the article match would be true as well, except when the expected answer is a chapter. Assume that the test data consists of these exact five cases, then the exact match score would be 2/5 = 40% and the article match score would be 3/4 = 75% (excluding the last case).

**Table 4.** Article Match and Exact Match Evaluation Example Cases

| Cases | Answer | | True/False | |
|---|---|---|---|---|
| | *Expected* | *Predicted* | *Exact Match* | *Article Match* |
| All Match | Article 2(1) | Article 2(1) | T | T |
| Incorrect Article | Article 3 | Article 4 | F | F |
| Correct Article, Incorrect Granularity | Article 3 | Article 3(1) | F | T |
| Correct Article, Incorrect Section/Letter | Article 5(1) | Article 5(2) | F | T |
| Chapter | Chapter 1 | Chapter 1 | T | N/A |
| **Scores** | | | **40%** | **75%** |

Aside from the correctness of the retrieved law component, we also evaluate the relevance of the answer itself. In some cases, the content of an article in a regulation is rewritten in several regulations. For instance, two or more regulations use the term "pekerja" (workers) and terms must be defined in the first article. Hence, some regulations contain the exact same text passage (or at least similar in text) for defining that term. Yet, the previous measures—exact match and article match—only consider the

law reference. To deal with such an issue, for each pair of expected and given answer, we also measure the F1-score of the word tokens of the pair, similar to the F1 measurements found in question answering tasks, including SQuAD [2].

For retrieval evaluation, we include mean average precision (MAP) measurement [13], [14]. Although we design the dataset in such a way that only one correct answer exists for each query, we consider the law components that are in the same granularity path to be relevant answers prior to reranking. For example, if the expected answer is Article 8 section (2), then the chapter (e.g., Chapter 2) and the article (Article 8) above it along with the letters in that section are considered as relevant answers. Given this set of relevant answers, we compare the query results with the set of relevant answers to measure the MAP.

## 3. Results and Discussion

We previously described how to develop our granularity-aware legal QA model. In this section, we elaborate and analyze the experimental results. The section is then followed by discussing the findings related to this research. Finally, we wrap up by elaborating on the possible future directions.

### 3.1. Abbreviations and Acronyms

We elaborate on the experimental results that are split into several aspects: BM25 feature impacts, retrieval results, end-to-end (retriever and reranker) results, and granularity accuracy analysis. We also give brief insights into these results and analysis.

#### 3.1.1. BM25 Feature Impact

Prior to testing different retriever models, we analyze the impact of additional features that can be incorporated to the BM25 ranking algorithm for retrieval. For this feature impact analysis, we use the multi-granularity indexing method where all the components on different levels of granularity are indexed. Based on the results in Table 5, question-to-statement conversion (Q2S) and chapter query expansion (Ch.) contribute to the incremental improvements in exact match and MAP scores. Using both of them (BM25 + Q2S + Ch.) yield the best results in Exact Match scores and Results in Top-k scores. On the other hand, simultaneous usage of text preprocessing techniques (stopwords removal and stemming) negatively impact the retrieval performance. The use of more formal language and consistent diction throughout legal documents is very likely to contribute to this degradation.

**Table 5.** BM25 Feature Impacts

| Retriever Model* | AM | EM | | QA F1 | MAP | Results in Top-*k* | | |
|---|---|---|---|---|---|---|---|---|
| | | *Micro* | *Macro* | | | *k=3* | *k=5* | *k=10* |
| BM25 (Baseline) | 58.28 | 21.62 | 18.89 | 35.99 | 42.41 | 36.76 | 49.73 | 63.24 |
| BM25 + Q2S | **59.51** | 24.86 | 21.90 | **37.75** | 42.93 | 38.92 | 49.73 | 63.24 |
| BM25 + Q2S + Ch. | 51.53 | **29.73** | **34.30** | 37.05 | 47.72 | **43.24** | **53.51** | **67.57** |
| BM25 + Q2S + Ch. + Stop. | 47.24 | 26.49 | 28.86 | 35.10 | 46.46 | 42.70 | 51.89 | 64.32 |
| BM25 + Q2S + Ch. + Stem. | 47.85 | 27.03 | 31.37 | 35.72 | **48.62** | 41.08 | 52.43 | 67.03 |
| BM25 + Q2S + Ch. + Stop. + Stem. | 45.40 | 27.03 | 31.37 | 34.76 | 47.31 | 41.26 | 52.43 | 64.86 |

[f] Q2S: question-to-statement; Ch.: chapter title query expansion; Stop.: stopwords removal (Indonesian); Stem.: stemming (Indonesian)

[g] Bold text: Highest value in a given column (evaluation metric or criteria).

Using question-to-statement conversion enables BM25 to better match the query with the expected answer by removing interrogative words and rearranging sentences into declarative sentences. Similar improvements can be found when implementing chapter title query expansion. This improves the performance for queries that expect the answer to be of chapter granularity. As a result, chapter components can be easily matched with the chapter title. This specific improvement in one granularity type can be identified by the improvement in exact match scores but with declining article match score (article match does not account for queries that are answered with a chapter).

We also further analyze as to how stopwords removal and stemming results in lower exact match scores by performing ablation study, i.e., adding each feature on top of the best result (BM25 + Q2S + Ch.). We find that adding stemming seems to be less detrimental than adding stopwords removal. In fact, adding stemming to the best result in exact match scores improves the MAP score. Although there are fewer exact matches found in the top-$k$, the relevant candidates in the same granularity path, tends to be ranked higher compared to BM25 + Q2S + Ch.

Since we aim to retrieve the expected component in the top-k, we prioritize the exact match scores and results in top-k over article match. Therefore, we consider BM25 with question-to-statement and chapter title query expansion (BM25 + Q2S + Ch.) to be the best result on this feature impact analysis. We will refer to multi-granularity BM25 as this best configuration in later parts of this section.

### 3.1.2. Retrieval Results

Next, we study how different retrievers affect the accuracy in retrieving the correct results. As a baseline we include the articles-only indexing method to represent the single-granularity retrieval as a weak baseline to be compared with multi-granularity retrieval. Based on Table 6, apart from the article match score, we find that multi-granularity BM25 has the best overall results and show a stronger performance over the Siamese models. The article match score is higher from the baseline than the multi-granularity since it contains fewer indexed documents while only evaluating on the article accuracy. However, the significant improvement by the multi-granularity BM25 shows that incorporating documents at different granularity levels contributes to the accuracy of the QA process.

**Table 6.** Retriever Results

| Retriever Model* | AM | EM | | QA F1 | MAP | Results in Top-*k* | | |
|---|---|---|---|---|---|---|---|---|
| | | Micro | Macro | | | *k=3* | *k=5* | *k=10* |
| BM25—Articles-only (Baseline) | **57.14** | 7.84 | 7.69 | 27.39 | 18.20 | 13.73 | 15.69 | 15.69 |
| BM25—Multi-granularity | 51.53 | **29.73** | **34.30** | **37.05** | **47.72** | **43.24** | **53.51** | **67.57** |
| Siamese (IndoLEM) | 24.54 | 4.86 | 3.94 | 16.79 | 9.81 | 11.35 | 17.30 | 21.62 |
| Siamese (IndoNLU) | 34.97 | 15.68 | 13.64 | 20.86 | 20.15 | 25.41 | 29.19 | 36.76 |

[h.] IndoLEM: indolem/indobert-base-uncased; IndoNLU: indobenchmark/indobert-base-p2.

[i.] Bold text: Highest value in a given column (evaluation metric or criteria).

### 3.1.3. End-to-End Results

We then evaluate the end-to-end results based on the retrieval and the reranking performance. For this evaluation, we use multi-granularity BM25 as the retriever with the best configuration (BM25 + Q2S + Ch.) as demonstrated in Table 5 and Table 6. We also try out different training methods, i.e., training objectives, for the reranking: relevance (R), granularity (G), and relevance + granularity (R+G). The results for the end-to-end experiment are shown in Table 7. We discover that most models can

answer around 30-36% of the questions with an exact match of the law component. The exact match score improvement over BM25 (with the best configuration) offered by the granularity-aware reranker reaches 20% relative to the baseline score. The best result based on having the most metrics with the highest score is by using Cross-Encoder on IndoNLU model, with 68.71% article match, 35.68% exact match, and 44.11% QA F1.

**Table 7.** End-to-End Results

| Reranker Model* | Training Method** | Top-*k* (Best) | AM | Reranked EM | | QA F1 | Isolated Reranking |
|---|---|---|---|---|---|---|---|
| | | | | *Micro* | *Macro* | | |
| BM25 Best Config. (Baseline; w/o reranking) | - | - | 51.53 | 29.73 | 34.30 | 37.05 | - |
| Siamese (IndoLEM) | R | 3 | 63.19 | 30.81 | 29.09 | 40.41 | 71.25 |
| Siamese (IndoNLU) | R | 3 | 63.19 | 30.81 | 29.85 | 40.12 | 71.25 |
| Cross-Enc. (IndoLEM) | R | 5 | 60.12 | 27.03 | 26.83 | 39.27 | 50.51 |
| Cross-Enc. (IndoNLU) | R | 5 | 66.87 | 30.27 | 29.21 | 43.05 | 56.57 |
| Siamese (IndoLEM) | G | 3 | 57.06 | 29.73 | 31.92 | 39.83 | 68.75 |
| Siamese (IndoNLU) | G | 3 | 60.12 | 31.89 | 33.16 | 39.58 | 73.75 |
| Cross-Enc. (IndoLEM) | G | 3 | 61.35 | 32.43 | 33.86 | 41.40 | 72.50 |
| Cross-Enc. (IndoNLU) | G | 3 | 62.58 | 34.59 | **37.40** | 40.99 | **80.00** |
| Siamese (IndoLEM) | R+G | 3 | 58.90 | 33.51 | 35.91 | 41.06 | 77.50 |
| Siamese (IndoNLU) | R+G | 10 | 63.19 | 34.59 | 34.09 | 42.83 | 52.03 |
| Cross-Enc. (IndoLEM) | R+G | 10 | 61.35 | **35.68** | 35.05 | 43.24 | **53.66*** ** |
| Cross-Enc. (IndoNLU) | R+G | 10 | **68.71** | **35.68** | 36.96 | **44.11** | **53.66*** ** |

[j.] IndoLEM: indolem/indobert-base-uncased; IndoNLU: indobenchmark/indobert-base-p2.

[k.] R: Performs reranking based on text relevance; G: Performs reranking based on granularity.

[l.] These isolated reranking scores are highlighted in R+G since these are the best scores for the given top-k parameter (k = 10).

[m.] Bold text: Highest value in a given column (evaluation metric or criteria).

Using different training methods contribute to the different values of *k* (of the top-k¬ retrieved documents) that yielded the best result for each reranker model. The reranker models that are trained only with only a single objective (G) and (R) prefer fewer documents to be retrieved as opposed to the ones trained for both granularity accuracy and document relevance objectives (G+R). Ultimately, the latter method tends to yield better results in the end-to-end evaluation.

We also find that reranking the granularity aspect (G) into training contributes to improving the results in general when compared to training the reranker only (R) to consider the relevance aspect. Yet, these performances can further be improved by combining both aspects (G+R) when training the reranker. Putting BM25 in tandem with these reranker trained for both objectives yields the best overall results for granularity-aware legal QA.

The isolated reranking scores are calculated by dividing the reranked exact match with the percentage of results that are in top-k. This is used in addition to the existing metrics to measure the reranking performance solely on the available documents for reranking. At its best, Cross-Encoder BERT can accurately rerank up to 80% of the cases with k=3. It is worth noting that higher k tends to result in lower isolated reranking scores.

### 3.1.4. Reranker Impact Analysis

We analyze how many test cases that are corrected by the reranker and how many that are instead ranked lower by the reranker. We group these test cases based on whether the answer is correct (exact

match) during the retrieval and the reranking steps and map them into the confusion matrix as shown in Fig. 4. We discover that the majority (37 and 41 cases) remain the same for retrieval and reranking. However, we find that there are more cases in which the reranker produces exact matches when the retrieval does not (29 vs. 18 cases). We further conclude that incorporating a granularity-aware reranker produces better answers overall.

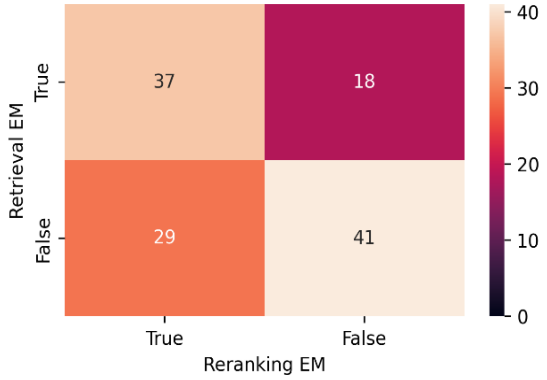


**Fig. 4.** Retrieval to Reranking Exact Match (EM) Confusion Matrix

### 3.1.5. Granularity Accuracy Analysis

We then analyze the granularity accuracy based on both the retrieval and the reranked results. Each group of predictions is mapped into a confusion matrix as shown in Fig. 5. These results indicate that the reranker manages to better distinguish the different granularities from the retrieved candidates with an accuracy improvement from 44.86% to 63.24%. The improvement is especially noticeable in the true positive values for letters, sections, and chapters. It further emphasizes the importance of incorporating the granularity aspect into ranking models when dealing with multi-granularity document corpora.

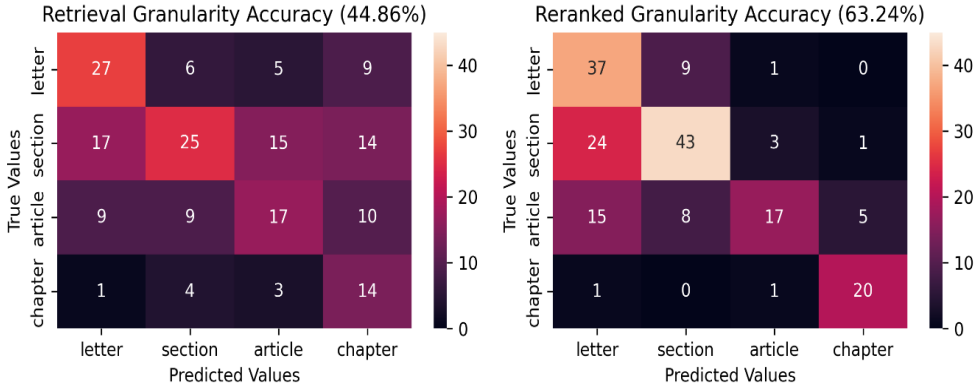


**Fig. 5.** Confusion Matrix of Granularity Accuracy for Retrieval (left) and Reranked (right) Results

Based on the different analyses that we do, we find that our model shows a significant improvement over the baseline. It improves the end-to-end results by 20% over BM25 on exact match scores. Then, analyzing the retriever and the reranker independently, we discover that the reranker increases the number of correct answer (relevance-wise and granularity-wise) and also increase the granularity accuracy to 63.24%.

We deploy the retrieve-rerank pipeline as a web interface as a HuggingFace Space. Additionally, our best reranker model (Cross-Encoder IndoNLU) is also available for reuse and transfer learning. We invite readers to test our system. Note that the system currently only supports Indonesian language.

### 3.2. Discussion

In this section, we highlight several points for discussion related to this study. We will touch on the multi-granularity information retrieval, semantic-based retrievers compared to BM25, and how the granularity-aware reranker can be leveraged for legal chatbots. Lastly, we also include a discussion related to the internal validity of this research.

#### 3.2.1. Multi-Granularity Information Retrieval

We have demonstrated multi-granularity information retrieval for regulation documents. However, this topic is not limited on laws and regulations, but also other structured documents. Other legal documents, e.g., multilateral agreements and code of conduct documents, are applicable for such a topic. It can even be expanded further for other structured documents, such as internal corporate documents to support their knowledge management systems.

In this study, we explore an approach to granularity-aware QA by leveraging multi-granularity retrieval and granularity-aware reranking. Other approaches are still open for future studies. Prior to proposing this approach, we performed preliminary experiments designing a pipeline using a granularity classifier when given a question. However, we decided not to continue with the approach due to underperforming preliminary results. Unlike intent classification in chatbots, granularity classification has an issue in which the granularity and the indexed documents are inherently dependent.

Benchmarking a multi-granularity information retrieval performance also poses a unique challenge, especially in hierarchical documents. For such a benchmark, we are required to have a certain measure of accuracy, not only in terms of the document relevance, but also in terms of the granularity accuracy. Our approach uses question-answering F1 score and article-match metrics in addition to exact match as less restrictive evaluation metrics. These metrics, however, do not have a definitive measure over distances between different granularity levels (e.g., should the distance between article-section and chapter-section be differentiated?).

Additionally, we previously performed granularity accuracy analysis with the confusion matrix displayed in Fig. 5. There are several errors in both granularity accuracy, mostly where letters are retrieved when sections or articles are expected. Upon further inspection, we find that most of these cases already retrieve the relevant result, but on the wrong granularity level. The model (especially BM25) seems to rank shorter documents higher, due to higher word token matches ratio over the length of the tokens (document). As for the reranked granularity accuracy, although chapters now contain less false positives after reranking, it appears that some sections and articles are still misclassified into letters.

#### 3.2.2. Semantic-Based Retrievers

In our experiments, we find that deep-learning-based retrievers are underperforming compared to the more traditional BM25. This may partly be attributed to the pretrained models used that are not trained for retrievals, but mainly due to legal documents are written in a formal language such that semantic search becomes detrimental to the query results. The formal language used in legal documents results in the terms used in those documents to be more consistent across passages and documents. The consistent use of terms helps to avoid ambiguity among legal experts [13]. Hence, the benefits of semantic similarity become less prominent over BM25. In general, there are other categories of techniques to measure text similarity for retrieval as pointed out in [42], such as knowledge-based and hybrid similarities.

#### 3.2.3. Leveraging Granularity-Aware Reranker for Legal Chatbots

The goal of this study is to propose a new task and model that can be leveraged to improve the answers' relevance produced by legal chatbots and virtual assistants. We find that the use of granularity-aware reranker improved the standard search results (BM25) by 20%. This helps in providing more precise answers over users' inquiries toward legal chatbots that are based on laws and regulations.

#### 3.2.4. Internal Validity

We identify a couple of potential selection bias which may influence the results of this experiment. The first one is the use of annotators that have a similar background in computer science, that may influence the dataset that has been generated. The second one is the modeling aspect, in which we only select pretrained models that is trained on Indonesian corpora and have published research supporting them. This approach avoids the exhaustive search of models, and therefore the result of this research is not meant to be interpreted as an ultimate benchmark of all models.

### 3.3. Future Work

In this section, we wrap up our discussion by pointing out several areas for future work related to granularity-aware legal QA. In a bigger picture, the task of granularity-aware QA can be generalized over hierarchically-organized documents, such as contracts, technical reports, manuals, and textbooks. Leveraging granularity-aware QA system over these types of documents can improve knowledge management systems that rely on these documents.

On the other hand, we are still expecting to see improvements in the modeling. Such techniques that can be explored are: multistage reranking and leveraging legal language models, such as LEGAL-BERT [25]. With the rapid advances in the field of Large Language Models (LLMs), our current model can be incorporated into a Retrieval-Augmented Generation (RAG) [43] pipeline in which our system acts as the retriever module and the LLM helps to summarize the answer and present it to the user.

## 4. Conclusion

In this research, we set out to improve the legal conversational AI experience, specifically on a chatbot's performance to answer users' questions related to legal domain more accurately. We chose to improve on the granularity accuracy of legal documents, an aspect that is novel for QA problems to be improved upon. To fulfill such a goal, we have proposed a novel task of Granularity-Aware Legal Question Answering, a legal QA task enhanced by adding the granularity accuracy over the text relevance objective in retrieving the correct law component. Along with proposing the task, we have generated a dataset for granularity-aware legal QA, consisting of 739 question-answer pairs over 15 government regulations regarding labor affairs in Indonesia. Finally, we have also designed and experimented with a retriever-reranker pipeline, the first system optimized for such a problem. We leverage BM25 as the retriever and BERT as the reranker with pretrained weights. By adding a granularity-aware reranker, we manage to improve the exact match results by 20% over a strong retriever baseline. We have discussed several points regarding the implications of this research and the future work. In practical context, our novel approach can help provide more precise answers, which is especially useful in legal chatbots. On the other hand, the concept can also be applied for other structured documents. Finally, our proposed model can further be incorporated into the state-of-the-art Retrieval-Augmented Generation (RAG) pipeline, giving much more natural interactions with the user.

## Declarations

**Author contribution.** All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper

**Funding statement.** This research was supported by a 2024 internal funding from the Faculty of Computer Science, Universitas Indonesia

**Conflict of interest.** The authors declare no conflict of interest.

**Additional information.** No additional information is available for this paper.

## Data and Software Availability Statements

We release our dataset used for training and testing the granularity legal QA pipeline consisting of 739 question-answer pairs over 15 Indonesian government regulations on labor affairs. The dataset is available on the following link: https://s.id/GranularityAwareLegalQADataset_Anon. We also make the reranker model and the interface accessible on Hugging Face: 1) Reranker Model: https://huggingface.co/kajix75/granularity-legal-reranker-cross-encoder-indobert-base-p2; 2) Retrieve-Rerank Interface: https://huggingface.co/spaces/kajix75/granularity-aware-indo-legal-qa).

## References

[1] W. Lehnert, "Human and Computational Question Answering*," *Cogn. Sci.*, vol. 1, no. 1, pp. 47–73, Jan. 1977, doi: 10.1207/s15516709cog0101_3.

[2] P. Rajpurkar, R. Jia, and P. Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, vol. 2, pp. 784–789, doi: 10.18653/v1/P18-2124.

[3] R. Karra and A. Lasfar, "Analysis of QA System Behavior against Context and Question Changes," *Int. Arab J. Inf. Technol.*, vol. 21, no. 2, pp. 191–200, 2024, doi: 10.34028/iajit/21/2/2.

[4] Z. Yang *et al.*, "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2369–2380, doi: 10.18653/v1/D18-1259.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North*, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[6] V. Karpukhin *et al.*, "Dense Passage Retrieval for Open-Domain Question Answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6769–6781, doi: 10.18653/v1/2020.emnlp-main.550.

[7] J. Moreno Schneider *et al.*, "Lynx: A knowledge-based AI service platform for content processing, enrichment and analysis for the legal domain," *Inf. Syst.*, vol. 106, p. 101966, May 2022, doi: 10.1016/j.is.2021.101966.

[8] M. Kaltenboeck, P. Boil, P. Verhoeven, C. Sageder, E. Montiel-Ponsoda, and P. Calleja-Ibáñez, "Using a Legal Knowledge Graph for Multilingual Compliance Services in Labor Law, Contract Management, and Geothermal Energy," in *Technologies and Applications for Big Data Value*, Cham: Springer International Publishing, 2022, pp. 253–271, doi: 10.1007/978-3-030-78307-5_12.

[9] V. Socatiyanurak *et al.*, "LAW-U: Legal Guidance Through Artificial Intelligence Chatbot for Sexual Violence Victims and Survivors," *IEEE Access*, vol. 9, pp. 131440–131461, 2021, doi: 10.1109/ACCESS.2021.3113172.

[10] D. R. Faisal, F. Darari, B. C. L. Tobing, and O. Lee, "Towards Building a Legal Virtual Assistant Based on Knowledge Graphs," *CEUR Workshop Proc.*, vol. 3257, pp. 73–78, 2022. [Online]. Available at: https://scholar.ui.ac.id/en/publications/towards-building-a-legal-virtual-assistant-based-on-knowledge-gra.

[11] M. Wyawahare, S. Roy, and S. Zanwar, "Generative vs Intent-based Chatbot for Judicial Advice," in *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, Mar. 2024, pp. 1–6, doi: 10.1109/IATMSI60426.2024.10502550.

[12] R. DALE, "Law and Word Order: NLP in Legal Tech," *Nat. Lang. Eng.*, vol. 25, no. 1, pp. 211–217, Jan. 2019, doi: 10.1017/S1351324918000475.

[13] J. Martinez-Gil, "A survey on legal question–answering systems," *Comput. Sci. Rev.*, vol. 48, p. 100552, May 2023, doi: 10.1016/j.cosrev.2023.100552.

[14] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. pp. 1-559, 2024. [Online]. Available at: https://web.stanford.edu/~jurafsky/slp3/.

[15] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval," *Introd. to Inf. Retr.*, Jul. pp. 1-461, 2008, doi: 10.1017/CBO9780511809071.

[16] S. Levy, K. Mo, W. Xiong, and W. Y. Wang, "Open-Domain Question-Answering for COVID-19 and Other Emergent Domains," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2021, pp. 259–266, doi: 10.18653/v1/2021.emnlp-demo.30.

[17] S. P. Widodo, "Comparative Analysis of Retriever and Reader for Open Domain Questions Answering on BPS Knowledge in Indonesian," *Proc. Int. Conf. Data Sci. Off. Stat.*, vol. 2023, no. 1, pp. 337–343, Dec. 2023, doi: 10.34123/icdsos.v2023i1.384.

[18] N. Abduljaleel, A. Corrada-Emmanuel, Q. Li, X. Liu, C. Wade, and J. Allan, "UMass at TREC 2003: HARD and QA," *TREC*, pp. 1–11, 2003, doi: 10.6028/NIST.SP.500-255.qa-umass.allan.

[19] J. Allan, B. Croft, A. Moffat, and M. Sanderson, "Frontiers, challenges, and opportunities for information retrieval," *ACM SIGIR Forum*, vol. 46, no. 1, pp. 2–32, May 2012, doi: 10.1145/2215676.2215678.

[20] F. Bu, X. Zhu, Y. Hao, and X. Zhu, "Function-Based Question Classification for General QA," no. 11. Association for Computational Linguistics, pp. 1119–1128, 2010. [Online]. Available at: https://aclanthology.org/D10-1109.

[21] V. Bolotova, V. Blinov, F. Scholer, W. B. Croft, and M. Sanderson, "A Non-Factoid Question-Answering Taxonomy," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2022, pp. 1196–1207, doi: 10.1145/3477495.3531926.

[22] A. Peñas *et al.*, "Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6241 LNCS, Springer, Berlin, Heidelberg, 2010, pp. 174–196, doi: 10.1007/978-3-642-15754-7_21.

[23] J. Rabelo, R. Goebel, M.-Y. Kim, Y. Kano, M. Yoshioka, and K. Satoh, "Overview and Discussion of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021," *Rev. Socionetwork Strateg.*, vol. 16, no. 1, pp. 111–133, Apr. 2022, doi: 10.1007/s12626-022-00105-z.

[24] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," 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)*, 2019, pp. 3980–3990, doi: 10.18653/v1/D19-1410.

[25] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets straight out of Law School," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 2898–2904, doi: 10.18653/v1/2020.findings-emnlp.261.

[26] S. Wehnert, V. Sudhi, S. Dureja, L. Kutty, S. Shahania, and E. W. De Luca, "Legal norm retrieval with variations of the bert model combined with TF-IDF vectorization," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, Jun. 2021, pp. 285–294, doi: 10.1145/3462757.3466104.

[27] B. Mansouri and R. Campos, *FALQU: Finding Answers to Legal Questions*, vol. 1, no. 1, pp. 1-4. Association for Computing Machinery, 2023. [Online]. Available at: https://arxiv.org/pdf/2304.05611.

[28] A. Askari, Z. Yang, Z. Ren, and S. Verberne, "Answer Retrieval in Legal Community Question Answering," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and*

*Lecture Notes in Bioinformatics)*, vol. 14610 LNCS, Springer, Cham, 2024, pp. 477–485, doi: 10.1007/978-3-031-56063-7_40.

[29] A. Hogan *et al.*, *Knowledge Graphs*. Cham: Springer International Publishing, pp. 1-237, 2022. [Online]. Available at: https://link.springer.com/10.1007/978-3-031-01918-0.

[30] S. Gao *et al.*, "How Legal Knowledge Graph Can Help Predict Charges for Legal Text," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14452 LNCS, Springer, Singapore, 2024, pp. 408–420, doi: 10.1007/978-981-99-8076-5_30.

[31] A. Revenko and P. Martín-Chozas, "Extraction and Semantic Representation of Domain-Specific Relations in Spanish Labour Law," *Proces. del Leng. Nat.*, vol. 69, pp. 105–116, 2022. [Online]. Available at: https://rua.ua.es/dspace/bitstream/10045/127407/1/PLN_69_09.pdf.

[32] J. S. Dhani, R. Bhatt, B. Ganesan, P. Sirohi, and V. Bhatnagar, "Similar Cases Recommendation using Legal Knowledge Graphs," in *SAIL'23: 3rd Symposium on Artificial Intelligence and Law*, Jul. 2021, pp. 1–12. [Online]. Available at: https://arxiv.org/abs/2107.04771v2.

[33] M. Abdurahman, F. Darari, H. Lesmana, M. Hartopo, I. Rhesa, and B. C. Lumban Tobing, "Lex2KG: Automatic Conversion of Legal Documents to Knowledge Graph," in *2021 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2021, pp. 1–5, doi: 10.1109/ICACSIS53237.2021.9631310.

[34] A. Hamid and Hasbullah, "Legal Hermeneutics of the Omnibus Law on Jobs Creation: A Case Study in Indonesia," *Beijing Law Rev.*, vol. 13, no. 03, pp. 449–476, Jul. 2022, doi: 10.4236/blr.2022.133028.

[35] G. Klyne, Jeremy J. Carroll, and B. McBride, *RDF 1.1 Concepts and Abstract Syntax*. pp. 1-22, 2014. [Online]. Available at: https://www.w3.org/TR/rdf11-concepts/.

[36] S. E. Robertson, S. Walker, S. Jones, and M. Hancock-Beaulieu, "Okapi at TREC-3.," *City*, no. January 1994, pp. 1–14, 1994, [Online]. Available at: https://www.researchgate.net/publication/221037764_Okapi_at_TREC-3.

[37] J. Bromley *et al.*, "Signature Verification using a 'Siamese' Time Delay Neural Network," in Advances in Neural Information Processing Systems, 1993, pp. 737–744. [Online]. Available: https://papers.neurips.cc/paper_files/paper/1993/file/288cc0ff022877bd3df94bc9360b9c5d-Paper.pdf.

[38] A. W. Pradana and M. Hayaty, "The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 3, pp. 375–380, Oct. 2019, doi: 10.22219/kinetik.v4i4.912.

[39] A. Liu, S. Swayamdipta, N. A. Smith, and Y. Choi, "WANLI: Worker and AI Collaboration for Natural Language Inference Dataset Creation," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Jan. 2022, pp. 6826–6847, doi: 10.18653/v1/2022.findings-emnlp.508.

[40] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 757–770, doi: 10.18653/v1/2020.coling-main.66.

[41] B. Wilie *et al.*, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020, pp. 843–857. [Online]. Available at: https://aclanthology.org/2020.aacl-main.85.

[42] D. D. Prasetya, A. Prasetya Wibawa, and T. Hirashima, "The performance of text similarity algorithms," *Int. J. Adv. Intell. Informatics*, vol. 4, no. 1, p. 63, Mar. 2018, doi: 10.26555/ijain.v4i1.152.

[43] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems*, 2020, pp. 1–16. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.