Jurnal E-Komtek
Vol. 9, No. 2. (2025) pp. 563-581
https://jurnal.politeknik-kebumen.ac.id/index.php/E-KOMTEK
p-ISSN : 2580-3719 e-ISSN : 2622-3066

Utilization of LLM in the Automation Process of Contract Template Recognition

Adam Ramdani Kusnandar[1], Herman Bedi Agtriadi[2]

[1-2] Department of Magister Computer Science, Institut Teknologi PLN, Indonesia, 11750

adam2430007@itpln.ac.id

https://doi.org/10.37339/e-komtek.v9i2.2511

Published by Politeknik Piksi Ganesha Indonesia

Artikel Info
Submitted:
23-06-2025
Revised:
08-12-2025
Accepted:
12-12-2025
Online first :
31-12-2025

Abstract

This research investigates the use of various text similarity methods in automating the recognition of varied contract templates. Determining the correct template is a crucial step before the automation process proceeds to the clause-by-clause evaluation stage. This recognition process involves dynamically comparing clause text between drafts and templates without data labeling, relying on available text. Testing was conducted using traditional methods (Jaccard similarity, TF-IDF, BM25) and natural language processing methods (BERT, LaBSE, LLM). The research methodology involves acquiring contract samples from various sources, creating templates, and testing template recognition. The testing output is evaluated based on its effectiveness in capturing semantic equivalence and contextual understanding. Research results show that LLM is highly robust in recognizing templates by only learning from the first few sample clauses. These findings indicate that template recognition automation through LLM will provide the best precision and accuracy compared to traditional methods and other natural language processing methods. Thus, this research can serve as a foundation for developing a template-based contract review automation system that is more robust against contract variations.

Keywords: *automation, contract analysis, clause similarity, large language models, legal analysis*

Abstrak

Penelitian ini menyelidiki pemanfaatan berbagai metode kesamaan teks dalam mengotomatisasi pengenalan template kontrak yang bervariasi. Penentuan template yang tepat merupakan langkah krusial sebelum proses automasi berlanjut ke tahap evaluasi pasal demi pasal. Proses pengenalan ini melibatkan mekanisme pembandingan teks klausul draft dan template secara dinamis tanpa dilakukan pelabelan data, dengan mengandalkan teks yang tersedia. Pengujian dilakukan dengan metode tradisional (Jaccard similarity, TF-IDF, BM25) dan metode pemrosesan bahasa alami (BERT, LaBSE, LLM). Metodologi penelitian melibatkan akuisisi sampel kontrak dari berbagai kalangan, pembuatan template, dan pengujian pengenalan template. Output pengujian dievaluasi berdasarkan efektivitasnya dalam menangkap kesetaraan semantik dan pemahaman kontekstual. Hasil penelitian menunjukkan bahwa LLM sangat robust dalam mengenali template hanya dengan mempelajari sampel beberapa pasal awal. Temuan ini menunjukkan bahwa proses automasi pengenalan template melalui LLM akan meningkatkan presisi dan akurasi terbaik dibandingkan dengan metode tradisional maupun metode pemrosesan bahasa alami lainnya. Dengan demikian, penelitian ini dapat menjadi fundamen bagi pengembangan sistem automasi review kontrak berbasis template yang lebih robust terhadap variasi kontrak.

Kata-kata kunci: *automasi, analisa kontrak, kemiripan klausul, model bahasa besar, analisis hukum*

This work is licensed under a Creative Commons Attribution-Noncommercial 4.0 International License.

1. Introduction

In recent years, the development of artificial intelligence (AI) technology has opened new opportunities for the advancement of automation across various sectors, including the legal field. Numerous studies have investigated the development of artificial intelligence targeting a range of legal activities [1], including legal research [2], [3], legal case analysis [4], [5], contract drafting/reviewing [6], [7], chatbots [8], automation [9], and others. Each of these activities possesses unique workflows characterized by their own dynamics.

Several legal activities, such as contract drafting and reviewing, are inherently repetitive and templated. Drafting or reviewing contracts typically relies on templates from previous similar contracts, with necessary modifications made as appropriate [10]. Templating becomes particularly crucial for institutions or companies engaged in repetitive and routine contract processes, such as procurement agreements, service agreements, subscription agreements, sales and purchase agreements, lease agreements, and so forth. These repetitive and routine processes occur largely due to substantive and even textual similarities. This offers ample scope for optimization in the drafting/reviewing process, particularly to enhance time efficiency, speed of completion, and consistency.

One of the major challenges in implementing artificial intelligence in the legal field is the uniqueness of legal language, especially in Indonesian. Indonesian legal language comprises a constellation of mutually characterizing relationships, is heavily logical (wordplay based on the correct logical rules of Indonesian), argumentative (raising foundations, bases, and reasons), and employs distinctive diction such as terms like *pleidoi* (plea), *tersangka* (suspect), *terdakwa* (defendant), *banding* (appeal), *memori banding* (memorandum of appeal), etc. [11]. In addition, the use of AI in the legal field has several limitations, such as a tendency toward oversimplification, ignoring critical information, failing to capture complex cause-and-effect relationships between rights and obligations, and lacking specific domain knowledge about contracts [12]. Given these limitations, AI is more appropriately used as an auxiliary tool that provides additional recommendations when making decisions [13].

To address these limitations, several studies have been conducted to facilitate the contract review process. Vectorization techniques from natural language processing and classification algorithms have been utilized to classify sentence types and identify related parties within contracts[14]. Furthermore, pipeline-based methods incorporating knowledge retrieval (text-

embedding-3-small), clause matching (BM25), and answer generation (gpt-4o-mini) have also been developed **[12]**. Interactive chatbot systems have been introduced to provide recommendations, contract templates, review guidelines, and annotations **[13]**. Large Language Models (LLMs) have become an essential part of the contract review process. Their abilities in reasoning, following instructions, and contextual understanding of probabilities, which can be enhanced through prompt engineering techniques **[15], [16],** are highly beneficial for further optimizing the use of LLMs in contract review processes.

In the contract review process, lawyers conduct analyses at the article-by-article level. This includes reviewing the standard clauses within each article, the relationships between articles, intra and inter-article risks, and an overall analysis of the contract. This workflow can be replicated using artificial intelligence, including LLMs. For LLMs to learn clause by clause, the crucial first step is the selection of an appropriate template to ensure that the standard clauses previously provided can serve as a benchmark for comparison against the clauses in the inputted contract draft.

This paper aims to explore the use of large language models (LLMs) in the automated recognition of agreement templates, an area that has not been previously examined. The focus is directed towards how LLMs can be employed to identify the structural and substantive conformity between draft agreements and relevant reference templates without resorting to NLP vectorization processes. This approach can serve as an important foundation for building legal automation systems that are simpler, more accurate, effective, and adaptable to the diversity of contract formats and substance, while preserving the authenticity of the text in both templates and drafts.

2. Method

The challenge encountered in the contract review automation process is how the automation system can precisely determine the appropriate contract template when a draft contract text is input, while also being able to identify when there is no suitable template due to insufficient similarity. To achieve accurate matching, it is necessary to perform sampling of the introductory text of the draft contract as well as the available template texts. The mechanism for matching these initial clause samples is conducted using traditional text similarity methods and natural language processing to generate a score that can determine which template should be selected. **Figure 1** illustrates the flowchart of the stages carried out in this study.

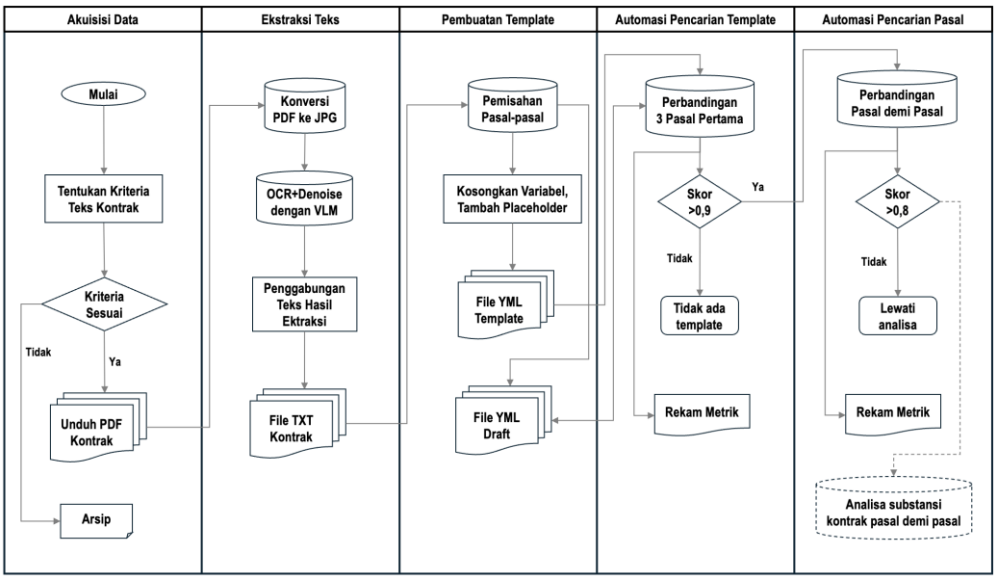

Figure 1. Template Recognition Automation Model Design

2.1. Contract Text Criteria

To ensure the variety of contracts obtained for the purpose of being used as templates and samples, the author established several criteria:

a. The contract text must be a final contract that has been signed by all parties involved, and not merely in the form of a draft or concept.
b. The source of the contract text should represent various sectors (government, businesses, public services, and academic institutions) in order to observe the standard sample patterns typically produced by each sector, with a minimum of two contracts from each category.
c. The contract text must be publicly available, considering that business contracts are generally confidential in nature.
d. It must be obtained directly from the official webpage of the respective entity.

The acquired contract texts are then reviewed to determine whether they can be properly extracted so as to produce complete contracts. In order to obtain high-quality templates, an examination is conducted of each clause in every extracted contract, with variable sections being omitted. This process also serves as a test for each method employed.

In bilingual contract texts, the template involves the removal of foreign language text. The aim is to ensure that only one Indonesian-language template serves as the reference source. This also serves as a test for the text similarity method used, to determine whether it can capture the semantic nuances between two distinct texts.

2.3. *Sample Collection*

Samples were collected in accordance with the four predetermined criteria. To this end, the author accessed the official websites of the Information and Documentation Management Officers (PPID) of the following entities, which provide information about executed contracts that can be directly downloaded. These contract documents are in various subjects, as follows:

1. Waste Management (2 documents)
2. Tourism and Creative Economy (3 documents)
3. Healthcare Services (9 documents)
4. Communication, Informatics, and Business (5 documents)
5. Religious Services (1 document)
6. Laboratory and Radiology Services (5 documents)
7. Educational Consulting Services (5 documents)

A total of 30 PDF archives containing contracts signed by the parties to the agreement were obtained from primary sources.

2.4. *Text Extraction Process*

The downloaded archives are non-searchable, image-based PDF files, which require the use of optical character recognition (OCR) technology for text extraction before being saved as text files. However, the OCR process presents its own challenges because many of the PDF files are scanned with low clarity, contain noise, and include irrelevant text such as page numbers, initial boxes, and "Scanned by" watermarks from mobile phone camera scans.

Therefore, an OCR mechanism is needed that can read text while also identifying which text should be excluded. The use of multimodal LLMs capable of understanding images containing text is required. This need is supported by the benchmark results related to the capacity of multimodal LLM models in reading text and performing OCR tasks, as detailed in **Table 1 [17]**.

Table 1. Benchmark OCR Multimodal LLM

Benchmark	Claude-3.5 Sonnet	GPT-4o	Qwen2-VL-72B
DocVQAtest	95.2	92.8	96.5
ChartQAtest	90.8	85.7	88.3
OCRBench	**788**	**736**	**877**

OCRBench was introduced to evaluate the capabilities of multimodal large language models (LLMs) in performing text recognition, scene-centric visual question answering (VQA),

document-oriented VQA, key information extraction, and handwritten mathematical expression recognition. OCRBench is designed to understand text in images with training data comprising a collection of 1000 manually filtered and corrected question-answer pairs related to text representation **[18]**.

As shown in **Table 1**, it can be observed that Qwen2-VL-72B possesses superior text-reading capabilities compared to the flagship paid models at the time, namely GPT-4o (OpenAI) and Claude-3.5-Sonnet (Anthropic), when tested on OCRBench. The Qwen2-VL-72B model is licensed under tongyi-qianwen, which permits both personal and commercial use **[19]**.

Based on these qualifications, the author selected Qwen2-VL-72B as the LLM model for contract text extraction purposes. By submitting the pages of PDF files in batch as JPG images, the pure contract texts resulting from the extraction are obtained.

2.5. Template Creation

At this stage, the extracted contract text is then sorted using a Python script designed to output text files containing individual articles in the YAML (YAML Ain't Markup Language) format. This format is a data serialization language designed to be human-friendly and to work efficiently with modern programming languages for common everyday tasks. Since this format is text-based, it is easy for humans to directly edit the text **[20].**

The next step is to manually edit each YAML file to clean up the extraction results from various variables within the clause text. Thus, we obtain clause texts that are easy to manage and can be refined as needed.

For bilingual contracts, at this stage only the Indonesian language texts are selected. In addition to the obligation under Article 31 paragraph (1) of Law Number 24 of 2009 concerning the National Flag, Language, and State Emblem, which requires every contract text to be in Indonesian, the interpretation of the contract will also prioritize the Indonesian language in the event of any differences in meaning with the other language.

2.6. Determination of Methods and Models

The determination of the similarity level between the draft text and the template employs both traditional methods (Jaccard, TF-IDF, BM25) and natural language processing (NLP) methods (BERT, LaBSE, and LLM). Traditional methods are utilized due to their speed and low latency, particularly in scenarios where parallel, large-scale pairwise text comparisons are

required. NLP methods are employed to ensure the comparison considers semantic context.

2.6.1. Jaccard Similarity

Jaccard similarity measures the similarity between two texts based on the ratio of shared words to the total number of unique words from both texts[21]. This approach is simple and fast; however, it is highly sensitive to editorial variations and, thus, is only effective when applied to texts with explicitly similar structures. Mathematically, Jaccard similarity is calculated using the following equation:

$$\boldsymbol{Jaccard}(\boldsymbol{A},\boldsymbol{B}) = \frac{|\boldsymbol{A} \cap \boldsymbol{B}|}{|\boldsymbol{A} \cup \boldsymbol{B}|}$$

For both the draft clause text and the template text, a conversion to lowercase and tokenization based on spaces is performed using `text.lower().split()`, and then wrapped in a set() so that Jaccard operates on unique elements. For example, tokenization is performed as follows:

- *Draft*: `{pihak, pertama, berhak, mengakhiri, perjanjian, ini, secara, sepihak, apabila, terjadi, pelanggaran}`
- *Template*: `{jika, terjadi, pelanggaran, maka, pihak, pertama, dapat, menghentikan, perjanjian, secara, sepihak}`

Thus, the intersection of the same words consists of 7 words: `{pihak, pertama, terjadi, pelanggaran, perjanjian, secara, sepihak}`. Subsequently, the union of all unique words is 15 words. Therefore, the Jaccard similarity score can be calculated as follows:

$$\frac{7}{15} = \mathbf{0,4666666667}$$

2.6.2. TF-IDF

TF-IDF measures the importance of a word in a sentence or text based on its frequency of occurrence and how common the word is across the entire text[14]. Although it does not account for semantic context, this method is capable of identifying clauses with distinctive terminology that frequently appears in contracts.

Texts are processed through tokenization using TfidfVectorizer, which breaks the text into word tokens, ignores punctuation, performs case normalization, and applies weighting. Both texts are represented as TF-IDF feature vectors. The similarity between vectors is calculated using cosine similarity with the following formula:

$$cosine\_similarity_{(A,B)} = \frac{A \cdot B}{||A|| * ||B||}$$

The similarity score will indicate the degree of overlap in significant words shared by both texts, taking into consideration the relevance of each word in the overall context.

2.6.3. BM25

BM25 measures text matching based on a probabilistic approach that calculates the relevance between compared texts, taking into account word frequency in the text as well as the length of the text relative to the average document length[22]. BM25 considers word distribution and document length, thus providing more relevant search results in the context of keyword-based matching, even though it still cannot comprehensively understand semantic meaning.

To obtain a similarity score using the BM25 method, the calculation is carried out in stages: (i) tokenizing the template and draft texts into words, (ii) calculating the frequency of appearance of template words in the draft, (iii) calculating the IDF for each word in the template text, (iv) then inputting the values into the BM25 formula for each word and summing them with the following formula[22]:

$$BM25(Q,P) = \sum_{\{i=1\}}^{n} IDF(t_i) \cdot \frac{[TF(t_i,P) \cdot (k+1)]}{[TF(t_i,P) + k \cdot (1 - b + b \cdot L(P))]}$$

2.6.4. BERT

BERT is used to measure contextual understanding of the compared texts in both directions (bidirectional) in capturing the meaning and nuances of words[23]. It not only recognizes identical words, but also understands the overall meaning of the sentences.

The similarity score between texts using the BERT method is obtained by inputting each text into the BERT model via an API interface to generate vector representations (embeddings), after which cosine similarity is calculated between the two embeddings. This approach is similar to the TF-IDF-based similarity computation method, where cosine similarity is also used to measure the closeness between text representations.

2.6.5. LaBSE

LaBSE (Language-agnostic BERT Sentence Embedding) is used to measure similarity and semantic nuances between sentences across different languages (multilingual)[24], enabling

resilience to text variations with different structures but similar meanings. In principle, the approach using the LaBSE model is the same as BERT, utilizing an API and cosine similarity calculation, but there will be differences in vector representations compared to BERT due to differences in training data and training techniques.

2.6.6. LLM

LLM is employed as a model expected to perform better in processing contract clause text comparisons. Specifically, the LLM used is AWQ quantized variant of Qwen3-8B, selected due to its availability on the Huggingface website[25], an Apache license with broad usage rights[26], as well as an aggregate benchmark score (MMMLU, MT-AIME24, PolyMath) that is comparable to GPT-4o[27].

The method for calculating the similarity score between the draft text and the template text using an LLM is quite simple, namely by relying on prompt engineering techniques as follows:

```
prompt = f"""
You are a helpful assistant that evaluates the similarity between two
legal texts. Text 1 (the draft) is in Indonesian and may contain
multiple languages. If Text 1 contains multiple languages, extract only
the Indonesian part before evaluating.
Text 2 is a legal template in Indonesian.

You will return ONLY a similarity score between 0 and 1, where 1 means
identical and 0 means completely different.
Ignore differences caused by unfilled blanks or variable placeholders.
Return only the number without any explanation.

Text 1 (draft):
{text1}

Text 2 (template):
{text2}

Evaluate the semantic similarity between these two legal texts and
return only a number between 0 and 1 (with 4 decimal places). Pay
attention to the main scope of both texts, as it is pivotal.

/no_think
"""
```

The prompt, which already contains both the draft and template texts, is then sent to the API providing the Qwen3-8B model. The output from the LLM will be in the form of a JSON file, which is then parsed, and only the output message containing the specified score is extracted.

2.7. Evaluation Process

Text reading is conducted on an article-by-article basis, with articles having been separated through the extraction process as previously described. This process is carried out in parallel to expedite processing time. The reading process is carried out in two stages: template retrieval and article-by-article retrieval within the template. Both stages are performed using the text similarity methods described in Section 2.6 above.

The mechanism for template and clause retrieval outlined below does not require a retrieval process that would increase latency and computational load. This retrieval process is designed to be highly adaptive to various templates, enabling its application to be customized for the specific business process scenarios of each entity.

2.7.1. Template Matching

To determine the type of agreement and its estimated substance, this can essentially be observed by reading several anchor clauses in the initial articles, which generally contain definition clauses and transaction clauses. The definition or general provision clauses are usually placed in Article 1, and the initial transaction clauses that follow (placed after the definitions) are articles concerning the purpose and objective, scope, and the subject of the agreement **[28], [29]**.

From the extracted texts—now forming the draft and template texts—this study selects the first three articles, which are considered sufficiently representative and decisive in text similarity comparison as well as to maintain processing speed. These draft articles are then compared against all available templates by reading the initial articles of each template. Template recognition is then carried out using a Python script that employs the six tested methods (Jaccard, TF-IDF, BM25, BERT, LaBSE, LLM) to yield output scores from the comparison between the draft texts and the available templates.

Measurement is conducted by referring to the commonly used confidence level, setting the figure of 0.90 as the minimum baseline to determine whether the chosen template is indeed the intended one. This is necessary due to the wide variation in text agreements, ranging from differences in letters to semantic differences in the clauses of a single article, which cannot be measured quantitatively. In the process of determining the similarity of the first three articles of the draft to the template to be compared, only the highest similarity score above 0.9 from the model's output is considered, to account for cases when more than one template is considered

a match by the applied method.

Through this approach, it can be determined how well the tested methods detect text similarity, both lexically and contextually. The more high scores found across multiple templates (i.e., low deviation among scores), the less capable a method is in distinguishing differences. Conversely, the higher the deviation of scores among the templates, the more precise the method is considered in discerning lexical or contextual differences.

At this template retrieval stage, each method produces a CSV file that records the category folder, template folder name, similarity score (Score_1), and label (Label = 1 for a score >0.9, Label = 0 for ≤0.9). The score is calculated based on the similarity between the template and the test data, and the label marks whether the result is deemed a valid template.

2.7.2. *Article-by-Article Matching*

After obtaining the appropriate template, the next process is to find the correct article for comparison. This involves comparing each draft article to every article within the template without requiring labeling. The automation system must be able to achieve accurate reading for this retrieval process.

Since this stage pertains to the substance of the clauses, it is crucial to achieve semantic similarity precision and insensitivity to variable differences such as placeholders or ellipses in the template that are filled in the draft. There may be instances where the draft contains extensive text, while the template consists of blank sections. Errors in selecting the correct article would result in inaccurate and even fatal analyses during contract review, leading the automation system to misinterpret the contract.

For the similarity threshold of clauses within the identified template, the author determines a value of 0.8, since each article within the template is likely to contain different provisions, but the threshold is not set too strictly to allow for more robust reading against lexical changes. Possible errors in the similarity comparison process may arise if the method ends up comparing a draft article against an incorrect article, as a text similarity method with poor discrimination might overgeneralize and fail to capture the differences accurately.

At the article-by-article retrieval stage, each method produces a CSV file that includes: the tested draft article, the determined template name from the template retrieval stage, the matched template article, the similarity score between the draft and template articles, the label (1 if the score >0.8, 0 if ≤0.8), and the ground truth (matching draft and template file names, i.e.,

both having the same file name in the extraction process).

3. Results and Discussion

3.1. Research Findings

3.1.1. Scoring of Template Matching Results

Testing the template retrieval process by comparing the first three articles of the draft against the first three articles of the template in each available template folder yielded the results as presented in **Table 2**.

Table 2. Template Matching Methods Similarity Scores

Method	Average Template Score	Average Non-Template Score	Template Matched	Non-Template Matched
Qwen3-8B	0.985036	0.079859	25	791
Jaccard	0.975706	0.107385	17	799
TF-IDF	0.989168	0.150238	22	794
BM25	0.997631	0.505977	32	784
LaBSE	0.972665	0.621096	31	785
BERT	0.934690	0.882458	700	116

There is a significant difference in the average non-template scores, which indicates the confidence level of each method in calculating text similarity. The number of template and non-template findings reflects how many of the templates analyzed fall within the >0.9 range (considered templates) or ≤0.9 (considered non-templates). However, in principle, this does not mean both are decided as the selected templates, since only one candidate template with the highest score is chosen.

Furthermore, it is necessary to obtain metrics that more clearly illustrate the confidence level of each method when comparing similarity or dissimilarity.

Table 3. Template Matching Confidence Score

Method	Min Score	Max Score	Range	Mean Score	False Positive
Qwen3-8B	0.0000	1,0000	1,0000	0.1076	0.12%
Jaccard	0.0320	1,0000	0,9558	0.1255	-0.86%
TF-IDF	0.0299	1,0000	0,9656	0.1729	-0.25%
BM25	0.1441	1,0000	0,8559	0.5253	0.98%
LaBSE	0.4767	1,0000	0,5235	0.6345	0.86%
BERT	0.8071	1,0000	0,1770	0.9273	82.84%

Table 3 presents a summary of the score distribution from each method. The range indicates the span between the lowest and highest similarity scores produced. A wider range implies greater discrimination capability between similar and dissimilar items. In this context, Qwen3-8B, Jaccard, and TF-IDF show full or near-full range usage, which may be useful in diverse datasets with varying degrees of similarity.

The mean score gives an idea of the general scoring tendency. Lower means like in Qwen3-8B (0.1076), Jaccard (0.1255), and TF-IDF (0.1729) suggest that these methods are more conservative and tend to score most comparisons as low-similarity. In contrast, BERT (0.9273) and LaBSE (0.6345) tend to produce higher similarity scores overall, potentially signaling reduced sensitivity to subtle differences.

The false positive rate is particularly insightful. A high false positive rate indicates that the method mistakenly identifies unrelated documents as similar. BERT, despite its high mean score, suffers from an extremely high false positive rate (82.84%), suggesting it tends to overestimate similarity. Meanwhile, Jaccard and TF-IDF show slightly negative false positive rates, possibly indicating an over-conservative threshold or rare cases of false negatives outweighing false positives. LaBSE and BM25 appear to maintain a good balance between semantic understanding and precision, with moderate mean scores and relatively low false positives (below 1%).

3.1.2. Scoring Article-by-Article Matching Results

From the results of matching each article in the draft against the articles in the predetermined template, a label value was obtained (1 if the score >0.8, 0 if ≤0.8). Subsequently, Accuracy, Precision, Recall, and F1-Score were calculated, with the results as shown in Table 4.

Table 4. Text Classification Metrics

Method	Accuracy	Precision	Recall	F1-Score
Qwen3-8B	0.9741	0.9740	1.0000	0.9868
Jaccard	0.9724	0.9700	1.0000	0.9847
TF-IDF	0.9657	0.9678	0.9967	0.9821
LaBSE	0.9549	0.9568	0.9977	0.9768
BERT	0.7901	0.7901	1.0000	0.8827
BM25	0.3045	0.3342	0.7625	0.4648

Accuracy reflects the proportion of correct predictions over the dataset. Precision

measures how many predicted positive results are truly relevant. Recall indicates how many actual relevant (positive) items were successfully identified. The F1-Score, as the harmonic mean of precision and recall, provides a balanced measure of performance.

Among all methods, Qwen3-8B achieves the best overall performance, accurately classifying nearly all instances with perfect recall and very high precision. Jaccard and TF-IDF, though based on traditional lexical similarity, also perform exceptionally well—surpassing both LaBSE and BERT in terms of F1-Score. While LaBSE still maintains strong performance, BERT, despite achieving perfect recall, suffers from lower precision, indicating a tendency to overpredict positives and thereby generating more false positives.

On the other hand, BM25 shows significantly lower performance across all metrics, especially precision and accuracy, suggesting that it frequently misclassifies dissimilar articles as similar. Nevertheless, it still retains a reasonably high recall, meaning it captures many of the actual positive cases, albeit with poor selectivity.

3.2. *Automation System*

The implementation of a contract review automation system based on templates can be further developed with more complex analytical systems utilizing more powerful Large Language Models (LLMs). As illustrated in Figures 2 and 3, precise recognition of templates is crucial and determines the success of subsequent processes, namely article-by-article comparison and substantive analysis of the articles. Files uploaded into the system are read, extracted, and then undergo a process to determine the appropriate template. Points 1 and 2 in **Figure 2** indicate that the corresponding template has been identified.

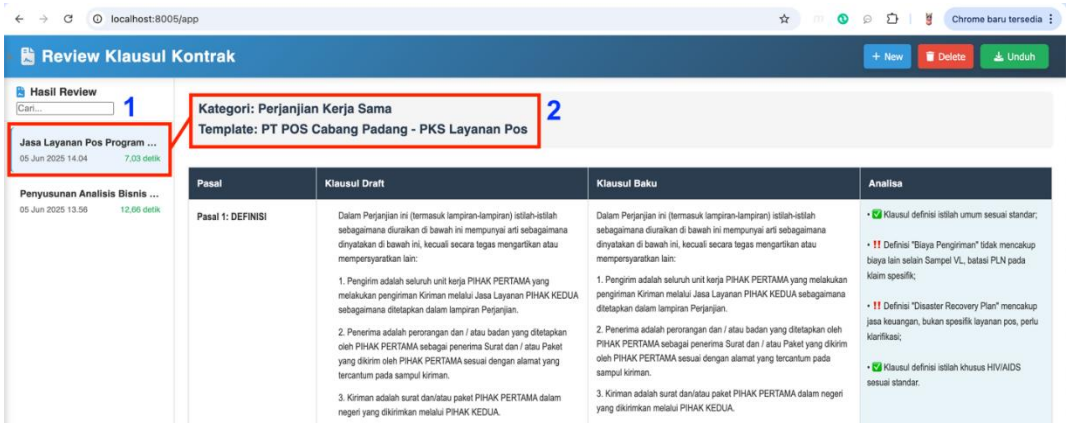

Figure 2. Contract Template Recognition Automation

In Figure 3, it is evident that the draft clauses match the template, which is critical for subsequent analysis. At point 3 in Figure 3, it can be seen that advanced analysis by the system can proceed once matching articles have been found. Point 4 illustrates that the template contains placeholders/blanks; the article-by-article recognition system must be robust enough to recognize and assess these as non-substantial differences. Point 5 also demonstrates that textual noise can still be properly handled by the clause recognition system.

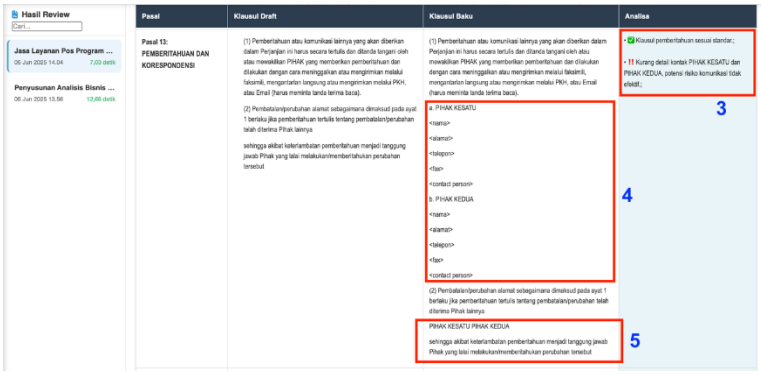

Figure 3. Article-by-Article Recognition Automation

3.3. Discussion

The high average template score indicates that each method considers the compared texts to be highly similar. Table 2 shows that the Qwen3-8B model exhibits refined similarity analysis with a wide score range, while the BERT model displays false positives, with the majority of its scores falling within the template score range. The lower the average score for non-template texts, the more significant differences the method recognizes among compared template texts. Conversely, a higher average score indicates that the method is overly confident in the similarity between templates. The precision of template selection greatly determines the overall subsequent article-by-article search process for further analysis.

The findings of this study suggest that the methodology for comparing contract clause texts can be effectively executed using an LLM with only 8 billion parameters (Qwen3-8B). This contrasts with prior research that relied on retrieval from embedding data[14] and traditional methods[12], which are acknowledged to oversimplification risk.

In Table 3, Qwen3-8B presents a minimum score of 0, demonstrating its ability to recognize texts that are entirely different at the article-by-article level. In contrast, BM25 records a minimum score of 0.8230, tending to assign high similarity scores. A wider score range signifies a method's

enhanced ability to distinguish substantial differences. Conversely, a narrower score range indicates higher generalization by the method.

The mean score implies that a lower value reflects the method's stricter criteria—less frequently assigning high similarity scores—whereas a higher mean indicates a more generalizing approach. Regarding standard deviation, a smaller value reflects highly stable and consistent confidence, but also points to the model's limited discernment of differences.

Furthermore, the coefficient of variation (CV) highlights that the LLM approach, such as Qwen3-8B, attains a higher ratio compared to BM25's very low ratio. This stability pattern suggests that methods with higher CV, like Qwen3-8B, yield a more diverse and dynamic distribution of confidence scores, potentially reflecting greater capability in distinguishing levels of certainty or offering richer score variation. In contrast, a lower CV value (as seen with BM25) shows a more uniform and less varied score distribution, indicating reduced sensitivity in distinguishing prediction confidence. Thus, within this context, a high CV is regarded as an indicator of more adaptive and informative confidence scoring.

The BERT and LaBSE methods demonstrate the ability to capture semantics, yet in most cases tend to generate false positives. This may occur due to insufficient training data or inadequate exposure to the Indonesian language. Moreover, semantic methods in BERT and LaBSE involve vectorization without direction, meaning the resulting vectors rely entirely on existing training data. This differs from the sensitivity of LLMs to prompting techniques, which allow for more directed outputs.

Article-by-article similarity matching is, in essence, a more straightforward process. Provided that the selected template candidate is correct, there tends to be substantive similarity, as reflected by high scores across most methods. BM25 often fails to recognize similarities in templates that frequently contain blanks marked by ellipses or placeholders, making these sections shorter than corresponding draft texts that already have inputted data and are thus longer.

In a template-based contract review automation system and article-by-article analytical context, text similarity methods play a vital role. As the number of templates increases, accurate template identification becomes crucial for maintaining accuracy. Errors in template selection can lead to mistakes in subsequent article-level analysis.

The advancement of contract review automation systems is anticipated to have a significant impact on increasing efficiency, effectiveness, and productivity, especially in organizations/companies with routine contractual documentation tasks. The speed at which LLMs process legal texts in seconds, as illustrated in Figures 2 and 3, demonstrates the tremendous potential for accelerating contract draft review among Indonesian legal practitioners. This research may serve as a foundation for improving the accuracy of LLM-based contract review processes.

Further research is warranted to evaluate other text similarity methods, including the use of transformer encoder models besides BERT/LaBSE or other transformer decoders aside from Qwen3-8B, particularly to achieve an optimal trade-off between accuracy and processing time.

4. Conclusion

This study demonstrates that Large Language Models (LLMs), even of modest size such as Qwen3-8B, can enhance the accuracy and efficiency of automation in template recognition and clause matching in Indonesian-language contracts. Unlike traditional approaches or semantic embedding models, LLMs have proven more adaptive to format variations and legal language nuances, resulting in more relevant and contextual analysis outcomes.

Therefore, the integration of LLMs into legal document automation workflows may constitute a breakthrough for the digitalization of legal services in Indonesia. This innovation paves the way for efficiency in large-scale document management and supports consistency in the analysis of repetitive legal review activities.

The findings of this study provide an important foundation for further research in the field of legal automation. It is hoped that subsequent research will explore other AI models and expand use-case scenarios so that Indonesia's legal automation systems become more reliable, inclusive, and better equipped to handle the diversity of documents in the future.

References

[1] K. Nitta and K. Satoh, "AI Applications to the Law Domain in Japan," *AsianJLS*, vol. 7, no. 3, pp. 471–494, Oct. 2020, doi: 10.1017/als.2020.35.

[2] Merine Thomas, "Quick Check: A Legal Research Recommendation System," , San Diego, US, Aug. 2020. [Online]. Available: https://ceur-ws.org/Vol-2645/short3.pdf

[3] Jhanvi Arora, Tanay Patankar, Alay Shah, and Shubham Joshi, "Artificial Intelligence as Legal Research Assistant," in *Forum for Information Retrieval Evaluation 2020*, Hyderabad, India, Dec. 2020.

[4] J. Drápal, H. Westermann, and J. Savelka, "Using Large Language Models to Support Thematic Analysis in Empirical Legal Studies," in *Frontiers in Artificial Intelligence and Applications*, G. Sileno, J. Spanakis, and G. Van Dijck, Eds., IOS Press, 2023. doi: 10.3233/FAIA230965.

[5] L. Karl Branting, "Automating Judicial Document Analysis," London, UK., Jun. 2017. [Online]. Available: https://ceur-ws.org/Vol-2143/paper2.pdf

[6] Kwok-Yan Lam, Victor C.W. Cheng, and Zee Kin Yeong, "Applying Large Language Models for Enhancing Contract Drafting," in *CEUR Workshop Proceedings*, Jun. 2023. [Online]. Available: https://ceur-ws.org/Vol-3423/paper7.pdf

[7] B. T. Wang, "Prompts and Large Language Models: A New Tool for Drafting, Reviewing and Interpreting Contracts?," *Law Tech Hum*, vol. 6, no. 2, pp. 88–106, Jul. 2024, doi: 10.5204/lthj.3483.

[8] D. Shu, H. Zhao, X. Liu, D. Demeter, M. Du, and Y. Zhang, "LawLLM: Law Large Language Model for the US Legal System," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, Boise ID USA: ACM, Oct. 2024, pp. 4882–4889. doi: 10.1145/3627673.3680020.

[9] M. Nithya, H. S, K. S, and S. K, "AI-Driven Legal Automation to Enhance Legal Processes with Natural Language Processing," in *2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)*, Bengaluru, India: IEEE, Dec. 2024, pp. 1246–1253. doi: 10.1109/ICICNIS64247.2024.10823316.

[10] Dan Simonson, Daniel Broderick, and Jonathan Herr, "The Extent of Repetition in Contract Language," Jun. 2019.

[11] S. S. Ola, "BAHASA INDONESIA RAGAM HUKUM," *Leksika*, vol. 3, pp. 37–43, Feb. 2009.

[12] E. W. Kim, Y. J. Shin, K. J. Kim, and S. Kwon, "Development of an Automated Construction Contract Review Framework Using Large Language Model and Domain Knowledge," *Buildings*, vol. 15, no. 6, p. 923, Mar. 2025, doi: 10.3390/buildings15060923.

[13] J. Zeng *et al.*, "ContractMind: Trust-calibration interaction design for AI contract review tools," *International Journal of Human-Computer Studies*, vol. 196, p. 103411, Feb. 2025, doi: 10.1016/j.ijhcs.2024.103411.

[14] I. Dikmen, G. Eken, H. Erol, and M. T. Birgonul, "Automated construction contract analysis for risk and responsibility assessment using natural language processing and machine learning," *Computers in Industry*, vol. 166, p. 104251, Apr. 2025, doi: 10.1016/j.compind.2025.104251.

[15] G. F. C. F. Almeida, J. L. Nunes, N. Engelmann, A. Wiegmann, and M. D. Araújo, "Exploring the psychology of LLMs' moral and legal reasoning," *Artificial Intelligence*, vol. 333, p. 104145, Aug. 2024, doi: 10.1016/j.artint.2024.104145.

[16] C. Zhang, J. Chen, J. Li, Y. Peng, and Z. Mao, "Large language models for human–robot interaction: A review," *Biomimetic Intelligence and Robotics*, vol. 3, no. 4, p. 100131, Dec. 2023, doi: 10.1016/j.birob.2023.100131.

[17] "Qwen/Qwen2-VL-72B-Instruct · Hugging Face." Accessed: Nov. 27, 2024. [Online]. Available: https://huggingface.co/Qwen/Qwen2-VL-72B-Instruct

[18] Y. Liu *et al.*, "OCRBench: On the Hidden Mystery of OCR in Large Multimodal Models," *Sci. China Inf. Sci.*, vol. 67, no. 12, p. 220102, Dec. 2024, doi: 10.1007/s11432-024-4235-6.

[19] "LICENSE · Qwen/Qwen2-VL-72B-Instruct at main." Accessed: Nov. 27, 2024. [Online]. Available: https://huggingface.co/Qwen/Qwen2-VL-72B-Instruct/blob/main/LICENSE
[20] Oren Ben-Kiki, Clark Evans, and Ingy, "YAML Ain't Markup Language (YAML™)," YAML Ain't Markup Language (YAML™) version 1.2. Accessed: Nov. 17, 2024. [Online]. Available: https://yaml.org/spec/1.2.2/
[21] S. Bag, S. K. Kumar, and M. K. Tiwari, "An efficient recommendation generation using relevant Jaccard similarity," *Information Sciences*, vol. 483, pp. 53–64, May 2019, doi: 10.1016/j.ins.2019.01.023.
[22] M. Kim and Y. Ko, "Multitask Fine-Tuning for Passage Re-Ranking Using BM25 and Pseudo Relevance Feedback," *IEEE Access*, vol. 10, pp. 54254–54262, 2022, doi: 10.1109/ACCESS.2022.3176894.
[23] S. Moon, S. Chi, and S.-B. Im, "Automated detection of contractual risk clauses from construction specifications using bidirectional encoder representations from transformers (BERT)," *Automation in Construction*, vol. 142, p. 104465, Oct. 2022, doi: 10.1016/j.autcon.2022.104465.
[24] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic BERT Sentence Embedding," 2020, *arXiv*. doi: 10.48550/ARXIV.2007.01852.
[25] Qwen Team, "Qwen/Qwen3-8B-AWQ," Qwen/Qwen3-8B-AWQ. Accessed: Nov. 27, 2024. [Online]. Available: https://huggingface.co/Qwen/Qwen3-8B-AWQ
[26] Apache, "Choose an open source license," Choose an open source license. Accessed: Nov. 27, 2024. [Online]. Available: https://choosealicense.com/licenses/apache-2.0/
[27] A. Yang *et al.*, "Qwen3 Technical Report," 2025, *arXiv*. doi: 10.48550/ARXIV.2505.09388.
[28] Salim H. S, *Hukum kontrak: teori dan teknik penyusunan kontrak*, Cet. 1. Jakarta: Sinar Grafika, 2003.
[29] Nanda Amalia, Ramziati, and Tri Widya Kurniasari, *Modul Praktek Kemahiran Hukum, Perancangan Kontrak*, Cetakan Pertama. Unimal Press, 2015.