DENSE RETRIEVAL SYSTEM FOR GENERAL COURT LAWS

Adriana NICOARA
M2 TAL
University of Lorraine
adriana.nicoara7@etu.univ-lorraine.fr

Dalila LADLI
M2 TAL
University of Lorraine
dalila.ladli9@etu.univ-lorraine.fr

Silviya SILWAL
M2 TAL
University of Lorraine
silviya.silwal2@etu.univ-lorraine.fr

ABSTRACT

There is a abundance of legal information available on the internet, but access to appropriate advice is always a challenge for those who lack domain knowledge. In this report, we present and discuss the development of a Civil Law dense information retrieval system made to identify a user's specific case and retrieve similar contexts from the available civil laws and case articles. Despite the lack of annotated data, we discover that our system does a fine job of retrieving relevant information based on user questions. According to the results, our Civile-Law-IR model outperforms other models that have been trained on millions of data points but have not been fine-tuned to domain knowledge.

1 Introduction

In today's age, the problem of accessing and understanding legal information is more pressing than ever and it gets reflected by the abundance of legal advice available online. However, it is a hindrance because laws and articles are intricately interconnected, making it easy for individuals who do not have professional assistance to become overwhelmed by the amount of information. On January 25th, 2019, the number of consolidated laws in France was 84,619 legislative articles and 233,048 regulatory articles, an increase from the previous year (2018), when the numbers were 83,254 and 231,363 respectively. This exponential increase presents a significant challenge to anyone attempting to understand a legal topic, whether professional or non-professional. To address this, tools have been created to assist in legal research by guiding users through their legal decision-making. In France, tools such as Dalloz [1] are primarily designed for professionals and employ legal jargon, making them difficult for users without legal knowledge to navigate and too expensive for non-professional use.

To address this issue, we attempted to develop a system that helps users in their legal research by providing information about similar cases, allowing them to skim through concrete examples and relevant articles. Our final system is a free and user-friendly solution which palliates the absence of legal education and awareness in society. The goal of this system is to aid in legal decision-making by providing quick access to important information that may impact legal outcomes. It is not intended to replace professionals such as lawyers or judges, but rather to empower individuals by increasing access to legal advice without the need for payment or legal action, reducing people's fear of asserting their rights.

Given the scarcity of legal assistance, we decided to focus on general cases from the French Court of Cassation, the highest court for all civil and criminal matters in the French judicial system. We selected cases from the Court of Cassation because its primary purpose is to ensure that the law is applied correctly in lower courts and to resolve any legal discrepancies. This makes the cases diverse and informative. The French Court of Cassation comprises six chambers: the First, Second, and Third Civil Chambers, the Social Chamber, the Commercial, Economic, and Financial

[1] https://www.dalloz.fr/

Chamber, and the Criminal Chamber. Each chamber specializes in a specific area, as their names suggest. We chose to utilize the Civil Chambers, as it offers the most diverse cases and generally deals with less complex matters.

This paper is organized as follows: in section 2 we describe few similar existing systems, in section 3 we describe our dataset, preprocessing steps and its use cases, in section 4 we describe in detail our system and evaluation methodologies, in section 5 we discuss the experiments, and in section 6 we discuss the results. Lastly, we also mention out front-end in section 7 some limitations and future work for our project in section 8.

2 Related Works

NLP techniques have been utilized in legal research, including in the study of Turkish law [1], where the authors used a variety of methods, such as decision trees, random forests, support vector machines, LSTM networks, and bidirectional LSTMs, with an attention mechanism for each model. They found that deep learning methods yielded the highest accuracy, with an average of 86.1%.

Additionally, an NLP model called Italian-legal-bert was developed for Italian law [2]. It is based on ITALIAN BERT XXL and was further trained on 235,000 civil cases. The model was evaluated by comparing it to Italian BERT on tasks such as identifying named entities by person type, measuring semantic similarity, and classifying rhetorical sentences by section class. The same approach was used for the French language, where the authors of JuriBERT adapted BERT models to the French language by pre-training them on French legal text, with the goal of helping legal experts [3].

In addition to models, open-source software packages have been created for NLP in law, such as LexNLP [4]. This package provides features such as document segmentation, information extraction, and the creation of unsupervised and supervised models like word embeddings and tagging models. It also includes pre-trained models based on unit tests drawn from documents in the SEC EDGAR database added to judicial and regulatory proceedings.

Furthermore, NLP models have been developed for the French language, such as BSARD [5]. This model includes over 1,100 French legal questions labeled by legal experts and linked to relevant articles from a corpus of more than 22,600 Belgian law articles. The authors utilized several retrieval approaches, including lexical and dense architectures, in both zero-shot and supervised settings. The results showed that fine-tuned dense retrieval models performed significantly better than other systems, with the best performing baseline achieving 74.8%. Our project shares a similar goal to BSARD, which is to make legal information more accessible to non-legal users, the only difference is that BSARD uses data from the Belgian legal system, making our project an innovation in the field of NLP applied to law.

3 Datasets

3.1 CASS Dataset

After exploring through multiple datasets, we decided to continue with the *CASS dataset*[2] as we thought it is the most appropriate for our task. This dataset contains the leading cases in judicial jurisprudence, specifically cases from the French Court of Cassation. It was updated with quite old data, civil chamber bulletin cases from 1960 and criminal chamber bulletin cases from 1963, up to cases from 2017. All the files present in the dataset are in XML format and contain the integral text of the case and summaries written by magistrates of the Court of Cassation.

Since XML format was not very readable, we converted all the files into a textual format with the help of a preprocessing script created especially for this dataset, from [6]. The new generated files are called *.story* file and the structure is: description of the case, followed by the token "@highlight", followed by the text of the summary. An example of a story file can be seen in Figure 1.

Because this dataset is so large, and because some laws may have changed since 1960, we decided to focus on and use only civil cases between 2000 and 2022, which totaled slightly more than 16000 files.

We made use of just the summaries from the dataset for two tasks:

- to compute the similarity between a user's question and summaries and retrieve k summaries
- to train a model for re-ranking the retrieved contexts

[2]https://www.data.gouv.fr/fr/datasets/cass/

```
 annulation , sur le pourvoi du sieur z... , d' un arret rendu , le 3 juin 1885 , par la cour de douai , au profit
la cour ,
oui m. le conseiller crepon , en son rapport ;
l' avocat du pourvoi , en ses observations ;
m. l' avocat general desjardins , dans ses conclusions , et apres en avoir delibere conformement a la loi ;
donne defaut contre le defendeur a la cassation ;
et sur les premier et deuxieme moyens du pourvoi ;
vu les articles 1121,1690 et 2075 du code civil ;
attendu , en droit , que le contrat d' assurances sur la vie , lorsque le benefice de l' assurance est stipule au p
que vainement on voudrait pretendre comme l' a fait l' arret attaque que dans un pareil contrat , l' assure ne stip
attendu , en effet que , d' une part , le profit de l' assurance peut , dans de certaines eventualites , revenir au
que , d' autre part , le stipulant s' engage a verser a la compagnie d' assurances des primes annuelles , de telle
attendu , conformement a la derniere partie de cet article que lorsque le tiers specialement designe par la police
attendu que la faillite du stipulant survenue avant son deces ne saurait faire disparaitre ce droit et autoriser le
attendu , d' autre part , qu' une police dans laquelle le benefice de l' assurance est stipule au profit d' un tier
que cet avenant laisse au contrat son caractere special de contrat d' assurance sur la vie , qui comporte , pour sa
attendu , en fait , qu' il resulte des constatations de l' arret attaque :
que , le 26 juin 1868 , le sieur x... a souscrit a la compagnie d' assurances generales une police d' assurance de
que , le 12 avril 1881 , par un avenant a la police de 1868 , le nom du sieur z... a ete substitue a celui de la fe
que dans cet avenant ont figure comme parties contractantes et ont signe : y... , le stipulant , le representant de
attendu qu' en l' etat de ces constatations , en decidant que la police n' avait cesse d' etre une creance dont le
par ces motifs , casse .
@highlight
le contrat d' assurances sur la vie , lorsque le benefice de l' assurance est stipule au profit d' une
personne determinee comporte essentiellement l' application des regles qui regissent la stipulation pour autrui .
de l' acceptation par ce tiers de la stipulation faite en sa faveur resulte pour lui un droit personnel et
irrevocable que la faillite du stipulant survenue avant son deces ne saurait faire disparaitre .
tant que cette acceptation n' a pas eu lieu , la police peut etre modifiee par un avenant substituant un autre
nom a celui qui avait ete primitivement inscrit . cet avenant laisse au contrat son caractere special de contrat
d' assurance sur la vie , qui comporte pour sa regularite l' intervention du stipulant et de la compagnie et qui
ne saurait etre confondu avec un contrat de transport dont la validite et les effets seraient subordonnes aux
significations prescrites par les articles 1690 et 2075 du code civil .
```

Figure 1: Generated .story file

3.2 Synthetic Question dataset

The generated synthetic questions were used to train the dense retrieval model for re-ranking, which is part of the dense retrieval model, explained in section 4.2. For each summary were generated one or more questions based on its length and complexity.

Before generating the questions, the data passed through some pre-processing steps. There were some inconsistencies and unwanted characters in the summaries that had to be eliminated, for example:

- Often, a sentence started in a row was finished in the next row so we eliminated all the characters for new line, in order to have continuous sentences. This was a little detail that we had to take care about, because it had a negative impact on the model used for generating question
- All words that contain an apostrophe were written with a white space after the apostrophe, for example *"n' est"* so we eliminated the white spaces
- In some summaries the character "\" was present with no reason so it had to be eliminated
- Sometimes some numbers were attached to the following word, making the meaning of the word unrecognizable for the computer. For example *"1l'arret"*; in these contexts, the numbers were eliminated.

For the question generation task we used the T5 Transformer model[3] that was fine-tuned in french for 3 different tasks: question generation, question answering, answer extraction. This model was fine-tuned on FQuAD[4] which is a french question answering dataset, with more than 25000 questions generated from wikipedia articles, and PIAF[5] dataset, also containing french question and answers. These both datasets were inspired by SQuAD[6] dataset.

All the questions together with their context were saved in a json file with the following format from Figure 2.

[3]https://huggingface.co/JDBN/t5-base-fr-qg-fquad

[4]https://fquad.illuin.tech/

[5]https://www.data.gouv.fr/en/datasets/piaf-le-dataset-francophone-de-questions-reponses/#description

[6]https://paperswithcode.com/dataset/squad

```
"JURITEXT000007040336.story": {
    "context": "ne constitue pas une contestation de la saisie au sens de l'article 66 du decret du 31 juillet
    "questions": [
        "Qu'est-ce qui ne constitue pas une contestation de la saisie au sens de l'article 66 du decret du 31
        "Qu'est-ce que la demande de saisie tendant à constater les manquements du tiers saisi?"
    ]
},
"JURITEXT000007040350.story": {
    "context": "viole l'article 7-a de la loi du 23 decembre 1986 la cour d'appel qui decide que le conge d'un
    "questions": [
        "Que peut faire le locataire d'un local qui a un usage professionnel?"
    ]
},
```


Figure 2: Generated questions

3.3 Civile-NLI dataset

For the purpose of creating the dense retrieval model (section 4.2) we use the synthetic Questions and context pairs to create a synthetic dataset. A typical NLI dataset contains premise, hypothesis, and labels. The labels are used to indicate the relationship between two given statements (premise and hypothesis). In most cases, the premise-hypothesis are labeled using three classes: entailment, contradiction and neutral. Entailment means that one statement logically follows from the other while Contradiction means that the two statements are mutually exclusive. Neutral means that the relationship between the two statements is uncertain or irrelevant.

In our case we are dealing with questions and context pairs for which we do not have the resource to annotated each pairs with the labels. To overcome this issue, we decided to automatically label the pairs using semantic similarity. The main heuristic behind creating this dataset is to map each synthetic question from the question generation task to the context it came from and label them as *related*. To create *unrelated* question-context pairs we used semantic similarity to find the least similar questions to each context and mapped these pairs as *unrelated* label. The algorithm behind this heuristic is described at Algorithm 1.

Algorithm 1: Synthetic NLI

Input: $contexts, questions$
Output: $nli\_dataset$
Function `Annotation`$(contexts, questions)$**:**
    $nli\_dataset \longleftarrow List()$
    foreach $(c, q) \in (contexts, questions)$ **do**
        nli_dataset.append(c, q, related)
        ce = embedding(c)
        qe = embedding(questions)
        matrix = cosine_similarity(ce, qe)
        matrix = sort(matrix)
        dissimilar = getTopk(matrix, questions, k=3)
        foreach $item \in dissimilar$ **do**
            nli_dataset.append(c, item, unrelated)
        end
    end
    return $nli\_dataset$
End Function

4 Methodology

Our approach is based on two main steps. First, we create a simple, semantic search model that retrieves from the summaries, the k most similar contexts to the user's question. The second step was to improve this semantic search model with a re-ranking model that returns the k most relevant contexts from the retrieved information.

4.1 Semantic Search Model

Semantic search is a type of search operation that aims to find items that are similar to a given question. This can be done by comparing the question to a set of items and ranking them based on their similarity to the question. Similarity can be determined using various methods such as cosine similarity, Euclidean distance, or Jaccard similarity, depending on the type of data being searched and the specific requirements of the application. Similarity search is often used in applications such as image and video retrieval, natural language processing, and recommendation systems, among others.

For the purpose of our project we created a semantic search model using cosine similarity and Facebook AI Similarity Search (FAISS)[7] as shown in Figure 3. To find the similarity scores between our user questions and contexts we use sentence embeddings. Sentence embeddings are a method of representing sentences as fixed-length real-number vectors. These embeddings capture the meaning of the sentence and can be used for information retrieval, and similarity measurements, among other natural language processing tasks. In order to get these sentence embeddings we use large pre-trained language models like Sentence-BERT [7]. This model takes a user question as input and retrieves relevant context based on the cosine similarity or FAISS. The model is setup for asymmetric semantic search which is is a type of semantic search that allows for different levels of similarity between the query and the results. In this type of search, the similarity measure used to compare the question to the items in the index is asymmetric, meaning that the similarity between the question and an item is not necessarily the same as the similarity between the item and the question. This allows for more flexibility in the search process and can produce more relevant results in certain cases.

For example, in a traditional symmetric semantic search, a question that contains the word “dog" would only return items that also contain the word “dog", but in an asymmetric search, it could return items that contain words such as “canine" or “puppy" as well. This is useful because it allows the search to take into account synonyms, paraphrases, and related terms, which can make the search more robust and improve the chances of finding relevant results. Asymmetric semantic search is often used in natural language processing and information retrieval applications, where the goal is to find items that are semantically similar to a given question rather than just syntactically similar.

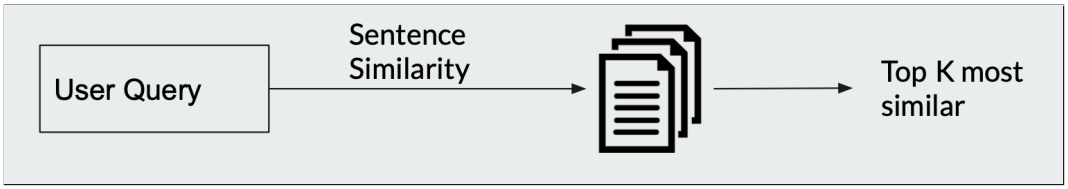

Figure 3: Semantic search system

4.2 Dense Retrieval Model

Dense retrieval models work by comparing the question’s embeddings to the context’s embeddings and ranking the items based on their similarity to the question. The difference between our semantic search model and the dense retrieval model is the added re-ranking component. The semantic search results are simply based on retrieval using similarity scores; however, these retrieved contexts are not completely relevant because the similarity score does not always reflect relevant information ranking. In ranked retrieval, the system returns an ordering over the context in the collection for a question rather than a set of context satisfying a question.

The re-ranking component for dense retrieval works based on natural language inference (NLI) models. In general, NLI models are trained to understand the semantic relationship between sentence pairs. In some research these models are trained to distinguish between entailment, contradiction, and neutral relationships. While in other cases, the sentence pairs can have continuous scores which define their similarity. NLI models can be used in dense retrieval systems to improve the relevance of search results by identifying semantically related documents or paragraphs, even if they do not contain the exact keywords queried. They can also be used to sort the results according to the strength of the semantic relationship.

For our dense retrieval system we improved on the semantic search model by training two natural language inference (NLI) models. We train the first model using the stsb-multi-mt french[8] dataset which is a publicly available dataset used

[7] https://faiss.ai

[8] https://huggingface.co/datasets/stsb_multi_mt/viewer/fr/train

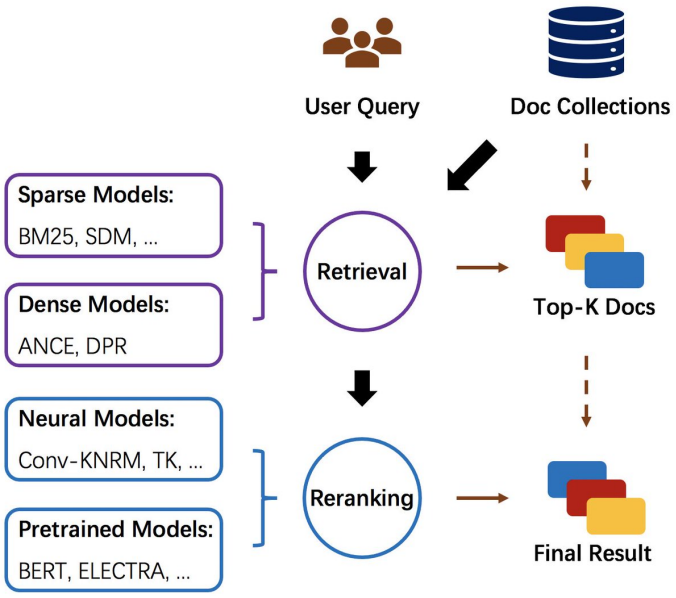

Figure 4: Basic architecture for Dense Retrieval Model

to train and test natural language inference models. It is a French version of the Sentences Involving Compositional Knowledge[9] (SICK) dataset that contains a set of sentence pairs with scores (continuous) indicating their similarity ranging from 0 - 5. The dataset contains a diverse set of sentences, allowing training models to handle a wide range of natural language phenomena. However, this dataset does not provide any domain specific training to our model which is why we created a synthetic nli dataset (3.3) using the synthetic questions and the contexts generated from the CASS dataset. Hence, we train another model using a subset of the Civile-NLI dataset. For the dense retrieval baseline model we use a pre-trained multilingual NLI model[10] In the end, we have two dense retieval model, and a baseline which have all been trained on different datasets.

The pipeline of our dense retrieval model is inspired from OpenMatch's Neu-IR research [8] which explains the two step pipeline for information retrieval shown in Fig 4.

4.3 Evaluation Methods

Because we lacked a gold dataset for evaluation, our first instinct was to conduct human evaluation. However, since human evaluation is costly and obtaining a sufficient number of participants is difficult, we also propose an automatic evaluation based on semantic similarity. Here, we describe our methodology for both evaluations.

4.3.1 Automatic Evaluation

Embedding based scores such as BERTscore [9] make use of token embeddings in the candidate sentence and reference sentence to compute similarity score for text generation. They find that BERTScore correlates better with human judgement scores. Based on this idea, we propose a similar evaluation metric where we compute the similarity between user question and the retrieved contexts using sentence embeddings. We compute the average cosine similarity between the question and each retrieved context while using a different sentence transformer[11] model unseen by our dense retrieval models.

4.3.2 Human Evaluation

Another method for assessing model performance is to use a rating scale of 1 to 5 (likert scale), where users rate the relevance of the models' output in relation to the question asked, with 1 being extremely irrelevant and 5 being very relevant. The idea behind our evaluation was to allow users to enter any civil law question and receive any number of

[9] https://paperswithcode.com/dataset/sick

[10] https://huggingface.co/amberoad/bert-multilingual-passage-reranking-msmarco.

[11] https://huggingface.co/nreimers/MiniLM-L6-H384-uncased

results. Users would then run the question against each model and rate the results. To avoid bias, the model names are not explicit in the interface, so users have no idea which models they are rating. The rating submissions are then saved in a json file, from which we can evaluate the models' performance based on their average rating score. However, because our evaluation system was built with Streamlit[13], collecting a significant amount of user feedback was difficult because deploying it on the cloud with GPU is expensive and running it on CPU is inefficient.

5 Experiments

DR-Baseline For the DR-baseline model we experimented with an existing pre-trained multilingual NLI model[10]. This model was trained on the Microsoft Machine Reading Comprehension Dataset (MS MARCO)[12] dataset which contains 400M tuples of a query, relevant and non-relevant passages. The training method of this model is similar to our STSB and Civile-law-IR mdoels. We use this model in a zero shot setting with no fine-tuning on our Civile-NLI-dataset.

STSB For the STSB model we made use of the STSB french dataset which provides pairs of sentences and a score of their similarity. Since this dataset contains continuous values for their similarity score, we train our model with a regression objective. While training, the sentence pairs are passed to the transformer network and a target value is predicted through a classification layer [10]. Hence, we used mean-squared-error loss as our objective function. For the transformer model we chose "camembert-base"[11] model which is a RoBERTa based model trained only on french data. The train set has 5749 sentence pairs, while the test set has 1379 sentence pairs. We train our model for 7 epochs with learning rate $2e-05$.

Civile-Law-IR In the case of Civile-Law-IR model we used the synthetic Civile-NLI dataset3.3. This dataset does not have continuous values but rather binary labels so we trained the model with a classification objective. Similar to the STSB model, given a question and context the model predicts the label using an additional classification layer on top of the transformer model. We used the same transformer model (camember-base) and cross entropy loss as our objective function. The Civile-NLI dataset contains 53K question and context pairs however, due to computational limitations we take a sample size of 10K for the train set and 1K for test set. We use the same hyperparameters as in the STSb model.

6 Results and Discussion

To get automatic evaluation results we made use of the generated questions (3.2) from which we selected questions not exposed to the model during training. We tested all three models, Civile-law-IR, STSB and DR-baseline model using the test set which contained 1K questions.

The results obtained from the automatic evaluation is presented in Table 1. The Civile-Law-IR model performs the best with 0.91 mean score followed closely by the DR-Baseline model. However, the STSB model does not perform as well as the others. This could be because of the training examples used for this model which does not contain any domain specific data. The sentence pairs in the dataset is also very short compared to the Civile-NLI dataset. Training the model with the STSB dataset seems to make the model perform worse than when no training is done (DR-Baseline model).

Furthermore, the DR-Baseline model performs almost on par with the Civile-Law-IR model, which can be attributed to the training datasets used by both models. Despite the fact that these models were trained on different domains, the datasets contained a similar structure of query and passage pairs. The Civile-Law-IR model was trained on only 10K of the available 53K training samples, whereas the DR-Baseline model was trained on 400M data points. It is worth noting that, despite the disparity in training data, the Civile-Law-IR model outperforms the baseline. When compared to the DR-Baseline model, increasing the training size for the Civile-Law-IR model may improve the score.

However, because we are averaging all of the semantic similarity scores using the same model, the ranking of the contexts is not taken into account in our automatic evaluation method. For example, suppose models A and B both return the same 100 contexts but model A has better context rankings than model B. In this case, using our evaluation method would return the same score for both models as we are averaging the score which does not take the rankings into account.

[12]https://microsoft.github.io/msmarco/

Models	Mean Similarity Score	Std.
Civile-Law-IR	0.91	0.03
STSB	0.87	0.05
DR-Baseline	0.90	0.04

Table 1: Overall Mean similarity score and Standard deviation on synthetic-nli dataset

To address this issue, we only consider the top 15 contexts from a total of 100 that are retrieved and re-ranked by each model. This ensures that, even if the top 100 contexts are the same across all models, each re-ranking model performs differently, avoiding the issue of having the same top contexts from each model. This is supported by the results in Table 1, where each model has a different score. Finally, we must keep in mind that if the top 15 contexts are the same but in different order, this evaluation metric will not reflect the models' ranking capability.

Models	Mean Similarity Score	Std.
Civile-Law-IR	0.88	0.018
STSB	0.83	0.022
DR-Baseline	0.86	0.019

Table 2: Overall Mean similarity score and Standard deviation on human generated questions

For the human evaluation method we consulted with a court bailiff, Mr. Hervé Pierson, who informed us that our results to some general questions are not very relevant. However, we performed an automatic evaluation using the same human generated questions and got the results as shown in Table 2. We observe that the semantic similarity scores are comparable to the ones obtained from evaluation done using synthetic questions. The contexts retrieved for these questions appear to have high semantic similarity to the questions, despite the fact that they appear to be irrelevant according to the court bailiff.

This contrast in evaluation could be explained by the main fields in our dataset. While pre-processing the CASS dataset we opted to only use data from the *Civil* folder to address general questions. However, this folder is divided into three divisions: The First Civil Division handles appeals related to civil status, family law, inheritance, contracts, nationality, literary and artistic property. The Second Civil Division deals with appeals related to insurance, civil procedure, and personal debt. The Third Civil Division handles appeals related to property and land registration. As a result, we realized that in order to fully assess the performance of our models, we needed to restrict our questions to specific fields within these Civil Divisions. We decided to forego human evaluations due to these issues and the additional limitations of GPU deployment of the evaluation system.

7 Front-end

To implement the front-end we used the Streamlit[13] python framework. This open-source framework is a very simple alternative for quickly creating a shareable web application without much front-end experience.

The graphical interface is simple and user friendly, as seen in Figure 5. The user can enter their question in the "Civil Legal Query" field and enter the number of results they want view in the "Choose Number of Result" field. Additionally, the user can choose between the three available models for which more information is provided via the "?" widget. Each of the three models is trained on different dataset, obtained in a different way as detailed in section 4.2. The system is deployed online on HuggingFace Spaces[14] but is running on the free CPU plan which means loading the results takes longer.

[13]https://docs.streamlit.io

[14]https://huggingface.co/spaces/ssilwal/CivileLaw-IR

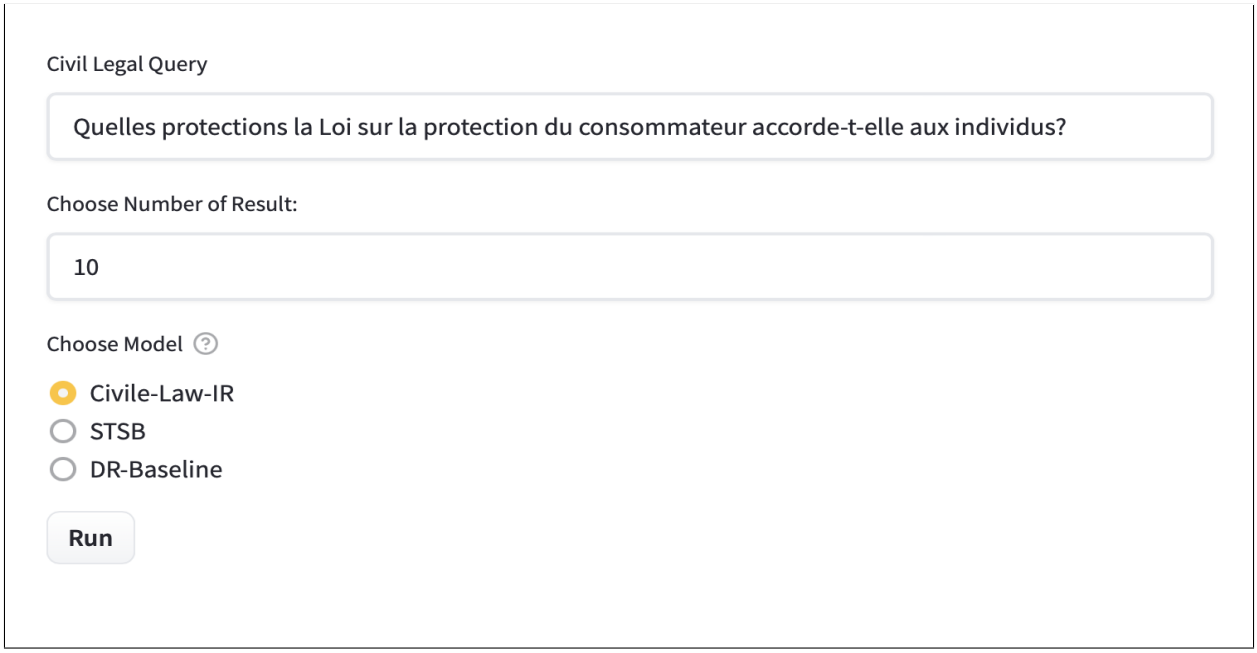

Figure 5: Front-end deployed on Huggingface Spaces

8 Limitations and Future Works

The main issue we encountered was the lack of computational resource for both training and deploying the front-end system, which resulted in less training data and slower performance. The lack of gold data for evaluation made assessing model performance difficult and hindering the fine-tuning process. Accessing gold annotated data is an expensive task because it requires the use of domain experts to provide annotation. We attempted to overcome this problem in our method by using synthetic data generation. Despite performing well in terms of semantic similarity, the system output is not relevant, according to expert advice. As a result, we recognize that our system is useful for dense retrieval tasks, but it requires a better annotated dataset to produce relevant search results.

Throughout the course of this project, we recognized the crucial role of dataset quality and quantity. The CASS dataset was extensive, but its scope was limited to specific topics and fields, with a focus on financial cases. As a result, users did not receive relevant answers to questions outside of financial issues. To improve our results, we would need to expand the dataset by partnering with government agencies, ministries or lawyer firms to legally obtain cases with a wider range of topics. A potential improvement for the system would be to employ a summarisation model which can simplify the documents before the retrieval and ranking process.

9 Conclusion

In this paper, we presented a solution for making legal knowledge available to the general public through a dense retrieval system without any annotated data. In terms of semantic similarity, our system performs well, but we emphasize the importance of using gold annotated datasets to improve performance. The limitations and challenges caused by a lack of data provide opportunities to experiment with synthetic data creation and evaluation; however, the results show that more experiments are required to deliver better performance.

References

[1] Emre Mumcuoğlu, Ceyhun E. Öztürk, Haldun M. Ozaktas, and Aykut Koç. Natural language processing in law: Prediction of outcomes in the higher courts of turkey. *Information Processing and Management*, 58(5):102684, 2021.

[2] Daniele Licari and Giovanni Comandè. Italian legal bert: A pre-trained transformer language model for italian law. volume 3256. CEUR, sep 2022. ISSN: 1613-0073.

[3] Stella Douka, Hadi Abdine, Michalis Vazirgiannis, Rajaa El Hamdani, and David Restrepo Amariles. Juribert: A masked-language model adaptation for french legal text. 2021.

[4] Michael J. Bommarito II, Daniel Martin Katz, and Eric M. Detterman. Chapter 11: Lexnlp: Natural language processing and information extraction for legal and regulatory texts. pages 216—-227, May 2021.

[5] Antoine Louis and Gerasimos Spanakis. A statutory article retrieval dataset in French. pages 6789–6803, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[6] Léo Bouscarrat, Antoine Bonnefoy, Thomas Peel, and Cécile Pereira. STRASS: A light and effective method for extractive summarization based on sentence embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics: Student Research Workshop*, pages 243–252, Florence, Italy, July 2019. Association for Computational Linguistics.

[7] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[8] Zhenghao Liu, Kaitao Zhang, Chenyan Xiong, Zhiyuan Liu, and Maosong Sun. Openmatch: An open source library for neu-ir research. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[9] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675, 2019.

[10] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019.

[11] Louis Martin, Benjamin Müller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. Camembert: a tasty french language model. *CoRR*, abs/1911.03894, 2019.

A Appendix

A.1 Repository

The link to our repository can be found here.

A.2 Acknowledgements

We would like to thank Mr. Hervé Pierson for his feedback during the evaluation of this project, as well as Miguel Couceiro and Esteban Marquer for their feedback sessions.