information

Article

An Intelligent Conversational Agent for the Legal Domain

Flora Amato [1], Mattia Fonisto [1,*], Marco Giacalone [2] and Carlo Sansone [1]

[1] Department of Electrical Engineering and Information Technology (DIETI), University of Naples Federico II, Via Claudio 21, 80125 Naples, Italy; flora.amato@unina.it (F.A.); carlo.sansone@unina.it (C.S.)
[2] Digitalisation and Access to Justice (DIKE), Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium; marco.giacalone@vub.be
* Correspondence: mattia.fonisto@unina.it

Abstract: An intelligent conversational agent for the legal domain is an AI-powered system that can communicate with users in natural language and provide legal advice or assistance. In this paper, we present CREA2, an agent designed to process legal concepts and be able to guide users on legal matters. The conversational agent can help users navigate legal procedures, understand legal jargon, and provide recommendations for legal action. The agent can also give suggestions helpful in drafting legal documents, such as contracts, leases, and notices. Additionally, conversational agents can help reduce the workload of legal professionals by handling routine legal tasks. CREA2, in particular, will guide the user in resolving disputes between people residing within the European Union, proposing solutions in controversies between two or more people who are contending over assets in a divorce, an inheritance, or the division of a company. The conversational agent can later be accessed through various channels, including messaging platforms, websites, and mobile applications. This paper presents a retrieval system that evaluates the similarity between a user's query and a given question. The system uses natural language processing (NLP) algorithms to interpret user input and associate responses by addressing the problem as a semantic search similar question retrieval. Although a common approach to question and answer (Q&A) retrieval is to create labelled Q&A pairs for training, we exploit an unsupervised information retrieval system in order to evaluate the similarity degree between a given query and a set of questions contained in the knowledge base. We used the recently proposed SBERT model for the evaluation of relevance. In the paper, we illustrate the effective design principles, the implemented details and the results of the conversational system and describe the experimental campaign carried out on it.

Keywords: legal AI; question and answer retrieval; intelligent user interface

Citation: Amato, F.; Fonisto, M.; Giacalone, M.; Sansone, C. An Intelligent Conversational Agent for the Legal Domain. *Information* **2023**, *14*, 307. https://doi.org/10.3390/info14060307

Academic Editor: Katsuhide Fujita

Received: 14 April 2023
Revised: 22 May 2023
Accepted: 24 May 2023
Published: 27 May 2023

1. Introduction

The legal domain has always been a challenging field of application for artificial intelligence techniques and ICT in general. The term "legal tech" refers specifically to the use of software systems to support the legal industry. During the last few years, text mining and natural language processing (NLP) technologies have significantly increased in the legal domain. A growing number of projects are leveraging machine learning (ML) models to extract useful information from legal documents.

Early approaches to NER for legal documents mainly relied on handcrafted rules and statistical learning models. In recent years, applications based on machine learning models and deep neural networks have become established. The most used applications of AI in the legal field are legal expert systems, i.e., computer applications able to imitate the process of consulting a legal expert to obtain specific advice for a given scenario. Since the law is a complex domain, many issue typologies are possible in developing legal expert systems. The first challenge is to give some weight to the principles of law but also to solving cases, the conclusions of which could improve the quality of the knowledge base. The second challenge is the nature of jurisprudence itself, as the law tends to be

flexible, it is not identically applied to the facts and the interpretation of jurisprudence may vary. Precedents should not be applied rigidly. Moreover, there are conflicting interests (e.g., dispute between the state and the individual) and the purpose of jurisprudence is also to balance them. Finally, it must be considered that different judges may emphasise different concepts. All this presents a degree of indeterminacy that is significant for the knowledge engineer. Another problem to be addressed is that legal expert systems are limited or geographically specific, each state has its laws and, obviously, a legal expert system implemented for a certain geographical area will not be valid elsewhere.

This work focuses on introducing an intelligent interface system [1] for the legal domain. The system was realised within the European project CREA2, which aims to provide a platform for the conflict resolution with equitative algorithms. The CREA2 platform will be equipped with an intelligent interface able to interact with users of different backgrounds—a chatbot [2,3]. The chatbot will guide the user within the platform, assist them in entering data, and explain all the offered functionalities [4,5]. The platform will offer dedicated services to users with different levels of legal knowledge. The bot must narrow the gap for less knowledgeable users on the subject when required. This innovative conversational user interface applies AI-driven tools based on machine learning [6] to implement innovative functionalities to help the practitioner, legal and regular users to set and progress the dispute resolution process.

An intelligent interface offers help to users of the CREA2 platform, a platform finalized for the application of a game theory algorithm in resolving disputes between people residing within the European Union [7]. Controversy is a dispute between two or more people who are contending over assets, in a divorce, an inheritance, or the division of a company. CREA2 will improve the existing CREA platform. The project will involve a wide range of EU stakeholders as the primary target groups, as well as lawyers, notaries, consumer associations, academics, students and policymakers. The CREA2 project builds on its predecessor's results, CREA [8] (2017–2019). The general project aims to introduce artificial intelligence (AI)-driven tools to assist lay and legal people in resolving their disputes by applying game-theoretical (GT) algorithms. The intelligent interface will help users to locate information of interest over a vast legal corpus [9] and offers support and suggestions, such as providing step-by-step guidance [10,11] in the dispute resolution process.

2. Related Works

Chatbots have become increasingly popular in recent years due to their ability to provide immediate and personalized assistance to users across a wide range of applications, such as customer support, sales automation, education, and even physical healthcare. Although chatbot technologies were first developed in the 1960s, they have experienced rapid growth thanks to advancements in machine learning and natural language understanding (NLU).

A chatbot, also known as a conversational agent, is an artificial intelligence (AI) software that can simulate a conversation (or a chat) with a user through text or voice interfaces [12]. Chatbots can process user inputs and generate appropriate responses using natural language processing and machine learning algorithms. The term "chatbot", short for "chatterbot", was initially coined by Michael Mauldin in 1994 to describe these conversational programs in his attempt to develop a Turing system [13]. Several techniques, approaches and technologies have been proposed in the literature for developing chatbots since the late 1990s. In the following, we describe the most common applications and use cases.

Chatbot systems are usually divided into four parts [14,15]: an interface, a multimedia processor, a multimodal input analysis, and a response generator. In detail:

1. the interface is responsible for managing the interaction between the chatbot and users, receiving inputs in various forms, such as text or audio, and returning appropriate responses;

2. the multimedia processor deals with voice or video signals and converts them into text or recognizes the user's tone to facilitate response generation;
3. the multimodal input analysis unit handles classification and data pre-treatment, often using NLU techniques, such as semantic parsing, slot filling, and intent identification;
4. the response generator associates a proper response to the given input taken from a stored dataset or maps the normalized input to the output using a pretrained model by using modern machine learning techniques.

The response generator is the core component of a chatbot where the question-and-answer process takes place. Based on the architecture of the response generator, chatbot systems can be classified into two main categories: retrieval-based chatbots, which select their responses from a predefined set of possible outcomes, and generative-based chatbots, which use ML techniques to generate answers [16] dynamically.

The goal of retrieval-based chatbots is to process the user input and choose the most suitable responses from a knowledge dataset. Four sub-categories of retrieval-based chatbots can be distinguished based on the architecture of their knowledge dataset and retrieval techniques. These categories are template-based, corpus-based, intent-based, and RL-based [17].

Template-based chatbots select responses from a set of possible candidates by comparing the user input to certain query patterns. This can be achieved using two main techniques:

- Pattern-matching algorithms: This is the most simple and oldest chatbot system, where answers are picked from a set of possible outputs based on keyword matching and minimal context identification. The first pattern-matching chatbot (and one of the first chatbots in history) was ELIZA [18], which, however, lacked the ability to maintain a conversation between humans and bots;
- Pattern-matching rules: In this case, pattern matching is executed using a set of scripts, engines or rules which define the chatbot behaviour. An important example is the artificial intelligence mark-up language (AIML), developed by Richard Wallace in 2003 for their chatbot A.L.I.C.E (Artificial Linguistic Internet Computer Entity), one of the oldest and most famous template-based chatbots [19]. At its core, an AIML script is an XML file made of different units called categories, represented by the `<category>` tag. AIML interpreters search through all categories one by one and return the `<template>` tag content whenever the user input satisfies some condition in the `<pattern>` tag, which supports wildcards and variables. A more recent example would be ChatScript, which is designed for interactive conversation and introduces more functionalities, such as semantic nets, logical conditions and functions. ChatScript won the 2010 Loebner Prize, fooling one of four human judges [20].

Although template-based chatbots have shown effectiveness in certain cases, their fundamental architecture necessitates scanning through all potential outputs for each input until the appropriate response is located. As a result, this approach can be slow and unsuitable for applications with a large knowledge dataset. To address this issue, corpus-based chatbots retrieve the appropriate response directly from a structured source (called the corpus) instead of relying on pattern-matching techniques. In this way, language tags and wildcards are not required, and the fetching process is more flexible, quick and scalable. This goal can be achieved differently, as reported below.

- Database-based corpus: Databases are the perfect tool to develop structured and organized knowledge, thanks to indexes and sorting algorithms. In this case, the response generator has to build a database query based on the normalized input and output the associated result. For instance, Pudner et al. [21] developed a method for generating an SQL query from relevant attributes and values from the user input;
- Semantic web-based corpus: An alternative data storage method involves using semantic webs. Unlike an SQL database, which consists of tables and rows, semantic webs employ semantic triples, which are comprised of three entities that encode data

in the form of subject-predicate-object. Consequently, data (subjects and objects) are stored as a set of entities connected directly by concepts (the predicates). Semantic webs typically use plain text files, facilitating their publication as web pages. Examples of chatbots that utilize semantic webs are detailed in [22,23]. In such instances, user inputs are transformed into SPARQL queries (the semantic query language used to retrieve data from a semantic web) to fetch the output response;

- Word-vector corpus: Using a technique known as word embedding, words and concepts can be stored as vectors [24]. This technique enables chatbots to calculate the metric distance between user inputs and query-response pairs, subsequently returning the result with the lowest distance.

Intent-based chatbots utilize machine learning techniques to establish a connection between user inputs and pre-defined outputs. Typically, relevant data is collected and stored to establish associations between user intents (i.e., the conceptual meaning behind a user's request) and appropriate responses. Next, a pretrained model leverages this information to link normalized user inputs with the most probable user intent [25]. Rasa is a well-known open-source machine learning framework used for automating text- and voice-based conversations [26]. Additionally, there are other notable examples, such as DialogFlow by Google [27] and wit.ai by Meta [28].

RL-based chatbots adopt reinforcement learning for response generation. In RL-based chatbots, each state s_i corresponds to a specific turn in the conversation and is usually represented by an embedded vector. After the chatbot is trained, it can select the most appropriate response (action) a_i to ensure that the conversation remains relevant and coherent [29].

Generative-based chatbots have the advantage of being able to generate responses dynamically, which can lead to more natural and flexible conversations with users. Generative chatbots can generate novel responses, which means that they are not limited to predefined responses like retrieval-based chatbots. This flexibility allows them to provide more personalized and relevant responses.

In terms of implementation, the most widely used approach until a few years ago was based on RNN (RNN-based chatbots). Now, however, many intelligent conversation systems are specifically based on transformers (transformer-based chatbots).

A transformer is a recent type of neural network architecture used for NLU and chatbots. First introduced in [30], transformers are also used in other tasks, such as language translation and text summarization. Transformers are based on the self-attention mechanism, which allows the model to learn which parts of the input sequence to attend to at each step of processing based on the relevance of the other parts of the sequence to the current position. This is performed through a process called scaled dot-product attention, where the model learns a set of weights to compute a weighted sum of the input sequence representations.

An important language model based on the transformer architecture is the generative pretrained transformer (GPT), which OpenAI developed in 2020 [31]. GPT serves as the underlying architecture for the ChatGPT chatbot, which has gained widespread recognition for its ability to provide detailed and articulate responses across various domains [32].

As discussed in this review of the current state-of-the-art in the chatbot landscape, the conversational agent sector is still very active and constantly undergoing new developments in its various fields of application, such as customer service, finance, healthcare, education, and personal entertainment. Both retrieval-based and generative-based chatbots have a wide range of applications: the former are more precise in their responses but less flexible in interpreting user questions, while the latter are more creative and diverse but less reliable in the accuracy of their responses. The advent of artificial intelligence and machine learning techniques will make chatbots increasingly efficient, helping with our daily tasks and accelerating our production processes [33].

3. Methodology

The goal of this chatbot is to appropriately respond to questions about a legal matter, which mainly involves providing valuable resources that could aid in resolving a legal conflict. Given a question about a specific legal conflict topic, we want to associate it with the proper answer for its resolution. This kind of task is called semantic search, an information retrieval task. Several attorneys and legal experts contributed to creating a dataset of tuples, structured as queries and answers, about different country-specific conflict resolution topics, which make up more than 300 entries. For each answer, multiple questions are provided, as shown in Table 1.

Table 1. Dataset entry example.

Question
• *How did Croatia implement the ADR EU Directive for civil and commercial matters, the ADR for Consumers EU Regulation, and the ODR Regulation at the national level?*
• *In what way has Croatia incorporated the ADR EU Directive for civil and commercial matters, the ADR for Consumers EU Regulation, and the ODR Regulation into its national laws?*
• *Can you explain how Croatia has implemented the ADR EU Directive, the ADR for Consumers EU Regulation, and the ODR Regulation at the national level for civil and commercial matters?*
Answer
The introduction of the Mediation Act in October 2003 was a significant step towards implementing alternative dispute resolution in Croatia. Since then, the use of alternative dispute resolution has gained popularity, and it has been recognized as a favorable alternative to the traditional judicial process. Croatia's Mediation Act, which was revised in February 2011, incorporates Directive 2008/52/EC of the European Parliament and of the Council of 21 May 2008 on certain aspects of mediation in civil and commercial matters. The Act governs mediation in various disputes, including civil, commercial, and labor disputes. In addition, Croatia has also implemented the Act on alternative dispute resolution for consumer disputes, which transposes Directive 2013/11/EU of the European Parliament and of the Council on alternative dispute resolution for consumer disputes.

To create such a bot, we evaluated several approaches:

One possible approach for such a problem is performing intent classification [34], a joint task within the NLP spectrum. It consists in identifying the intent behind a given query, which, in our case, also involves associating it with a proper answer. This task requires a large dataset of annotated examples for the model to be accurate. Since our dataset is slight, such an approach may not be helpful.

The subsequent approaches overcome the data scarcity problem by leveraging the similarities between the queries and the corpus of documents to identify the most relevant response to a given question. Instead of learning how to classify an intent from a never-read query, it retains a function of the degree of difference (similarity function) between two queries to mimic intent classification. It exploits the theory behind few-shot learning [35], a deep-learning approach that allows statistical models to perform accurately on new data with minimal training examples through similarity functions.

Such an approach is question-and-answer retrieval [36], which consists in finding the most relevant answer from a large corpus of answers to a given question. This task requires a large corpus of possible answers (or documents) to be effective. Since our corpus is small, such an approach may not be functional.

The last approach is similar question retrieval, which involves finding questions semantically similar to a given one from a corpus of available questions. Instead of retrieving the most relevant answer to the query, we try to find the most pertinent question from our dataset that is semantically similar to the query. We can then associate that with its related answer, just as we did with intents and answers for intent classification. This approach can be practical in cases where the dataset of question and answer is small, we do not have a large corpus for answers, and we have several query formulations.

Since this best suits our scenario, we adopt the last approach by exploiting the SBERT models, while not fully investigating the others due to time and data availability constraints.

3.1. Sentence-BERT

Sentence-BERT (SBERT) is a modification of the BERT network using Siamese and triplet networks that can derive semantically meaningful sentence embeddings with minimum computational overhead. While BERT is designed to generate contextualized representations of words, SBERT is specifically designed to create representations for entire sentences or paragraphs. For instance, finding the most similar pair in a collection of 10,000 sentences requires about 65 h of inference computations with BERT, while only 5 s with SBERT. This enables BERT to be used for specific new tasks, which did not apply to BERT before. These tasks include large-scale semantic similarity comparison, clustering, and information retrieval via semantic search.

An SBERT model adds a max pooling operation to the output of BERT to derive a fixed-sized sentence embedding, and then its BERT component is fine-tuned, creating Siamese and triplet networks to update the weights such that the produced sentence embeddings are semantically meaningful and can be compared with a similarity function [37].

In our case, we took advantage of the semantic search potential of the SBERT architecture to improve search accuracy by understanding the content of the query. As depicted in Figure 1, at inference, an SBERT model takes two sentences A and B, computes their semantically meaningful numerical representations u and v and then compares them by means of a similarity function, such as cosine similarity. This is especially intriguing for this application since a human may ask a question in several different ways. While the words may differ, the queries may still have semantically similar meanings.

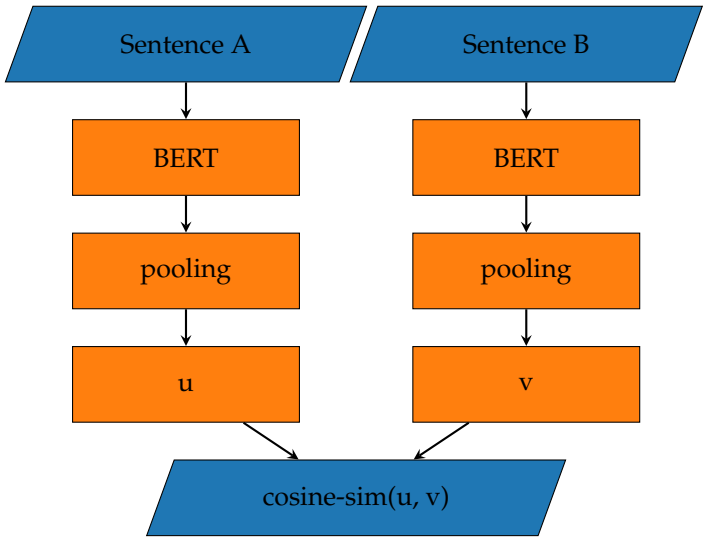


Figure 1. SBERT architecture at inference to compute similarity scores.

3.2. Semantic Search and Similar Question Retrieval

The following mathematically describes the semantic search algorithm:

Given a search query q and a set of documents $D = \{d_1, d_2, \ldots, d_n\}$, the semantic search algorithm finds the highest semantically similar documents in D to the query q. More precisely, it ranks each document d_i based on a similarity score. SBERT models can create embeddings for queries and documents within the same high-dimensional vector space, such that two sentences close within the high-dimensional vector space are semantically similar.

Let $F(\cdot)$ denote the embedding vector obtained by applying the SBERT model over a sentence. The similarity between a sentence x and a document y can be calculated using the cosine similarity as follows:

$$semsim(x, y) = cossim(F(x), F(y)) = \frac{F(x) \cdot F(y)}{\|x\|\|y\|}$$

where *semsim* represents the semantic similarity function between two sentences, *cossim* the cosine similarity function between two vectors, $\cdot$ the dot product and $\|\cdot\|$ the L2 norm. Note that *cossim* can be replaced by any other similarity function between two vectors, such as the dot product.

Given the embedded query $F(q)$ and the embeddings $F(d_i)$ for all the documents in the corpus D, semantic search computes the similarity score as the cosine similarity between the query embedding and each document embedding. Then these documents are sorted in descending order according to their similarity score to determine which documents are the most relevant to the query based on some predefined threshold value. A semantic search prediction for a query q will be considered correct if its relevant document d within the training set entry (q, d) has a similarity score greater than the threshold. In this way, we can exploit classification metrics based on true/false positives and negatives, such as precision and recall.

Similar question retrieval is a semantic search where the query q is a question and the set of documents D comprises other questions. As for the Quora duplicate questions problem [38], the user enters a question, and the algorithm retrieves the most similar questions from the dataset using the semantic search logic. The SBERT model for this task is trained to identify similar questions over a dataset of question-question pairs. This is a symmetric search task as the search queries have the same length and content as the questions in the corpus. After an appropriate question representative has been found, the associated answer within the corpus is returned [39], as shown in Figure 2.

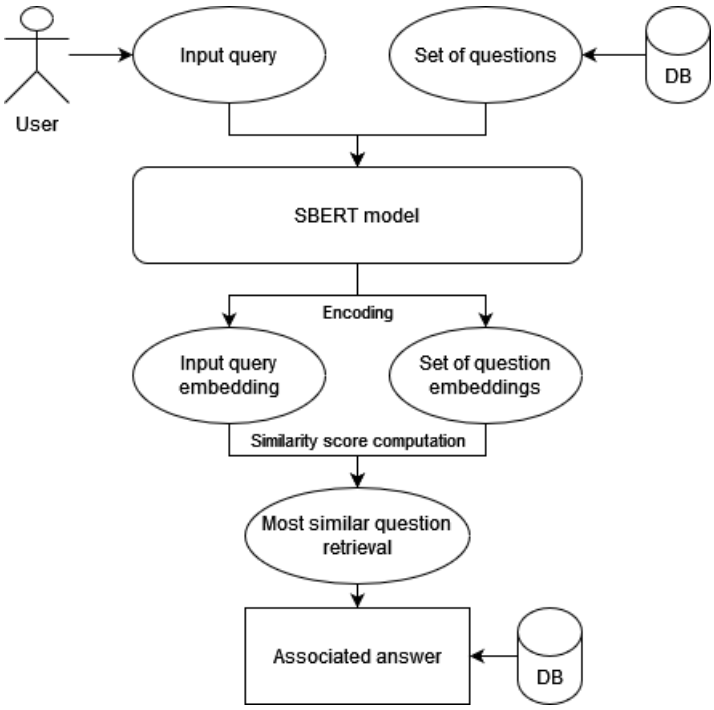


Figure 2. Flow diagram illustrating the adopted methodology at inference.

3.3. Data Augmentation

When building a chatbot, one of the critical tasks is adding training data for all intents to cover. To ensure an assistant can handle real-world conversations, we need to anticipate different variations in which users can express the same intent. The more varied training data we add, the hope is that the natural language understanding (NLU) model becomes more robust to incoming user queries. There is a need to have data augmentation, which helps in producing these variations to augment the training data with various paraphrases. Using a paraphrasing model, which takes as input a sentence, we generate multiple paraphrased versions of the same sentence.

Parrot [40] is a paraphrase-based utterance augmentation framework purpose-built to accelerate training NLU models. It is based on T5, namely, the Text-to-Text Transfer Transformer [41], which is a transformer-based language model that operates on a "text-to-text" framework, where it is trained to convert input text from one form to another. It is pretrained on a large corpus of diverse text data and fine-tuned on specific downstream tasks using a unified objective of maximum likelihood estimation. T5 exhibits strong performance across a wide range of natural language processing tasks, including text classification, summarization, translation, and question answering, by treating these tasks as text-to-text transformations.

The Parrot model fine-tunes the T5 model on the paraphrasing task. It generates paraphrases for a given query that convey the same meaning, are grammatically correct, and as different as possible on the surface lexical form by means of the Levenshtein distance. The first two characteristics, namely, adequacy and fluency, are preserved and can be easily controlled with this framework by means of thresholds. After the T5 component of the Parrot framework outputs the paraphrases, two models are used for estimating the adequacy and the fluency: first, the input query along with the paraphrases are passed as input to an adequacy language model that estimates for each possible tuple an adequacy score between 0 and 1. According to the set threshold over adequacy, only the ones that have a higher score than the threshold are preserved. Then, these selected paraphrases are passed as input to a fluency language model that estimates for each paraphrase a fluency score between 0 and 1. According to the set threshold over the fluency, only the ones that have a higher score than the threshold are preserved. Lastly, the Levenshtein distance is computed for each possible tuple consisting of the input query and a selected paraphrase to estimate their diversity and sorted in descending order according to such distance.

Since each intent is made of a few different queries, we first take one out for each in the original dataset to generate the test set. The training set now consists of 220 questions (intents) having, on average, 1.28 different formulations each. Then, we augment the training queries using this framework with an adequacy threshold of 0.95 and a fluency threshold of 0.90, generating on average 5.40 different formulations for each intent and obtaining a 4.22 times larger training dataset. Since the likelihood of an augmented question in the training set resulting in the same question in the test set is not zero, we double-checked that there are not any exact matches between the two sets. The data is saved in a JSON format, where each entry contains a list of questions and a corresponding answer, with each question and answer represented as separate strings. The idea behind not augmenting prior to the test set generation is not to mix training and test data in any way, to avoid the network outputting misleading performance metrics. An example of query augmentation is shown in Table 2.

Table 2. Augmented query example

Question
Can you explain how Croatia has implemented the ADR EU Directive, the ADR for Consumers EU Regulation, and the ODR Regulation at the national level for civil and commercial matters?
Paraphrases
• *What steps has Croatia taken to implement the ADR for the Consumers EU Directive and the ODR Regulation?*
• *How does Croatia implement the EU Directive on dispute resolution and the ODR regulation at national level?*

4. Results

All the experiments were conducted on a workstation equipped with a Ryzen 5 7600X processor, 32GB of DDR5 RAM, and an RTX 3070 graphics card with 8GB of VRAM. We fine-tune and evaluate two pre-trained SBERT models for this task: an `all-mpnet-base-v2` trained on a vast dataset tuned for many use-cases and a `quora-distilbert-multilingual` trained to identify similar questions over the Quora duplicate questions dataset. These two models are, respectively, based on MPNet [42] and DistilBERT [43]: the former is a multi-task pretrained neural network model that combines the training objectives of masked language modeling (MLM) and translation language modeling (TLM) to learn a shared representation of multilingual text, and the latter is a smaller and faster version of the BERT model that is distilled from the original BERT model by retaining its most essential weights through a teacher-student training process. We evaluate the performances of the two over a test set of 1 query per intent: we first compute the overall accuracy of the pretrained models as is, then we fine-tune them and compute the overall accuracy again for comparison. A prediction for the test question t_i, the one that belongs to the intent i-th, will be considered correct if the training question with the maximum similarity score over all the training questions q_j is given by a question within the same intent, such that $i = j$.

During the fine-tuning process, we compute some evaluation statistics by means of a binary classification evaluator, which evaluates a model based on the similarity of the embeddings by calculating the accuracy of identifying similar and dissimilar questions. The evaluator computes the embeddings for all questions and evaluates the pair-wise cosine-similarity between all possible pairs, then marks all questions with a cosine similarity higher than a certain threshold as semantically similar. However, as it does not know if it should mark all questions with a similarity above 0.5 as semantically similar, or, perhaps, with a score above 0.75, it determines the optimal cosine similarity threshold for each epoch and computes the statistics.

The optimal threshold is determined by iteratively examining the cosine similarity scores and similarity labels (similar/dissimilar) of question embeddings. Starting with the highest cosine similarity score, the evaluator progressively moves to lower scores. For each score, it calculates the precision, recall, and F1-score based on the number of positive examined instances and the total examined instances. The threshold is then set as the average between the current score and the next score in the sorted list. The evaluation metric we are interested in is the F1-score, the harmonic mean of precision and recall. This process continues until the highest F1-score is found. The interpretation of this threshold is related to the model's confidence in assessing two semantically similar questions, while the F1-score represents the model's overall performance in distinguishing between similar and dissimilar questions. The dataset for this evaluator, as well as the fine-tuning process, is made of all the possible triplets (a_i, p_i, n_j), where a_i is the anchor test question of the i-th intent, p_i is a positive semantically similar training question of the same intent, and n_j is a negative semantically similar training question of a different intent, such that $i \neq j$.

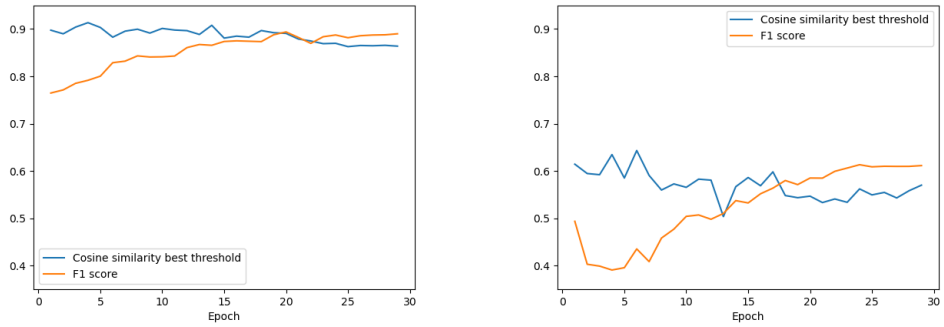
Fine-Tuned Models Comparison

We refer to the same training configuration for the fine-tuning process for both networks in Table 3.

Table 3. Fine-tuning training parameters

base model	`quora-distilbert-multilingual` or `all-mpnet-base-v2`
training data	augmented training question triplets (q_i, p_i, n_j)
batch size	16
epochs	30
data reshuffling	at every epoch
warmup steps	first 10% of training steps
loss function	BatchHardTripletLoss [44]
evaluator	BinaryClassificationEvaluator
evaluation data	augmented test question triplets (t_i, p_i, n_j)

Thanks to the evaluator, the F1-score with the associated optimal cosine similarity threshold is computed for each epoch over the evaluation data and plotted in Figure 3.



(**a**) `quora-distilbert-multilingual` (**b**) `all-mpnet-base-v2`

Figure 3. Model evaluation results.

The F1-score for the `quora-distilbert-multilingual` keeps fiercely increasing from 0.70 on the first epoch to 0.89 on the last epoch, while the optimal cosine similarity threshold keeps slightly decreasing from 0.94 on the first epoch to 0.88 on the last epoch. The model is somewhat less confident about the semantic similarity between the legal-specific questions but its performances are much better than before fine-tuning.

On the other hand, the F1-score for the `all-mpnet-base-v2` keeps slightly increasing from 0.49 on the first epoch to 0.61 on the last epoch, while the optimal cosine similarity threshold keeps slightly decreasing from 0.61 on the first epoch to 0.57 on the last epoch. The model is somewhat less confident about the semantic similarity between the legal-specific questions and its performances are much worse than before fine-tuning.

By just looking at the raw evaluation metrics, the `quora-distilbert-multilingual` looks promising for identifying semantically similar legal-specific questions, while the `all-mpnet-base-v2` is the opposite. Note that the best F1-score of the latter is worse than the worst of the former.

We also compare them with a metric compatible with their specific task: identifying the correct intent for each question. For these configurations, we compute the overall accuracy over the test set and plot them in a heatmap fashion in Figure 4.

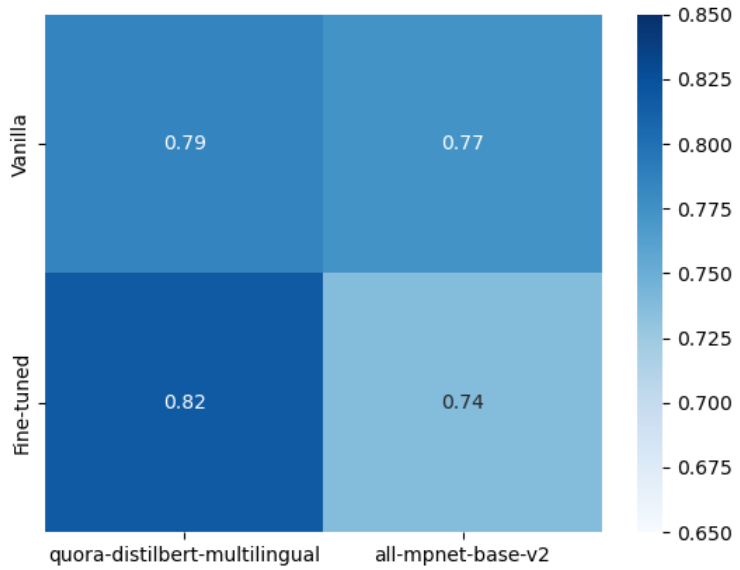


Figure 4. Model accuracies comparison.

As expected, the fine-tuned `quora-distilbert-multilingual` model reaches a 0.82 accuracy, outperforming the fine-tuned `all-mpnet-base-v2` model with a 0.74 accuracy. The latter also worsens its accuracy when fine-tuned. Overall, both models perform quite nicely on the classification task over the test set, reaching a high accuracy.

5. Discussion

We have successfully built a deep-learning-based information retrieval system using a dataset of legal conflict resolution question-answer pairs that can be used as a foundation for a conversational agent to provide valuable resources when prompted with similar queries. We leveraged SBERT models to associate each question with the related answer more appropriately with the related intent, addressing the problem as a similar question retrieval instead of a question-and-answer retrieval. We fine-tuned two pretrained models and evaluated their performances using the F1-score during training and intent classification accuracy after fine-tuning over a set of test questions. The fine-tuned `quora-distilbert-multilingual` model outperformed the other one, most probably because it was pretrained on a similar task over the Quora dataset, specifically for similar question retrieval.

Several challenges and limitations were highlighted: most of them due to the dataset size. In fact, it is worth noting that a question-answer approach may have been more effective if provided with a large corpus of possible answers. It would have led to a more comprehensive understanding of legal conflict resolution questions and potentially improved the overall system's performance. But given the limited size of our dataset, a question-question approach was more appropriate. In addition to this, it would have been valuable to explore the system's performance on a broader range of legal topics and investigate ways to enhance its generalization capabilities, since our solution only takes into account a specific set of legal topics. Finally, human evaluation as an evaluation metric could have been considered to provide a more exhaustive assessment. Overall, this work has shown the effectiveness of SBERT models for building an information retrieval system able to provide useful legal conflict resolution answers to common queries.

Future directions of this work envision a broader solution to providing legal advice involving a generative chatbot powered by a large language model (LLM) and a database of useful legislative instruments and legal cases. It would comprise of two main components: a "solution explorer" component and a "digital journey" component. The former leverages the power of embedding language models, such as Sentence-BERT (SBERT) or similar models, to retrieve documents, such as legislative instruments and legal cases, relevant

to the user query within a database. These can later be exploited by the LLM to support the informed response. The latter employs an LLM, such as Llama [45] or similar models, fine-tuned on a vast corpus of legal texts along with a reasoning framework, such as the Reason+Act (ReAct) framework [46] or similar frameworks, which provide a structured approach to generating responses by guiding the LLM reasoning process, to generate responses able to provide guidance for conflict resolution through the reasoning. In essence, this work, with appropriate modifications, can be a foundation for developing the "solution explorer" component of the envisioned future work.

Author Contributions: Conceptualization, F.A., M.F., M.G. and C.S.; Data curation, F.A. and M.G.; Formal analysis, M.F.; Funding acquisition, F.A. and M.G.; Investigation, F.A. and M.F.; Methodology, F.A., M.F. and C.S.; Project administration, F.A. and M.G.; Resources, F.A. and M.G.; Software, M.F.; Supervision, F.A. and C.S.; Validation, F.A., M.F., M.G. and C.S.; Visualization, F.A. and M.F.; Writing—original draft, F.A. and M.F.; Writing—review & editing, F.A., M.F., M.G. and C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the European Union grant number 101046629—CREA2. However, the views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

Data Availability Statement: Restrictions apply to the availability of these data. Data are confidential and available only to the members of the consortium CREA, a European project co-funded by the Justice program 2014-2020, call JUST-AG-2016-05, under grant agreement no. 766463.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Hassani, H.; Silva, E.S. The role of ChatGPT in data science: How ai-assisted conversational interfaces are revolutionizing the field. *Big Data Cogn. Comput.* **2023**, *7*, 62. [CrossRef]
2. Setlur, V.; Tory, M. How do you converse with an analytical chatbot? Revisiting gricean maxims for designing analytical conversational behavior. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 29 April–5 May 2022; pp. 1–17.
3. Ferreira, A.; Gama, T.; Oliveira, F. Construction of an Instrument for the Quantitative Assessment of Experience in the Use of Conversational Agents. In Proceedings of the HCI International 2022-Late Breaking Papers. Design, User Experience and Interaction: 24th International Conference on Human-Computer Interaction, HCII 2022, Virtual Event, 26 June–1 July 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 226–243.
4. Parviainen, J.; Rantala, J. Chatbot breakthrough in the 2020s? An ethical reflection on the trend of automated consultations in health care. *Med. Health Care Philos.* **2022**, *25*, 61–71. [CrossRef] [PubMed]
5. Ng, J.; Haller, E.; Murray, A. The ethical chatbot: A viable solution to socio-legal issues. *Altern. Law J.* **2022**, *47*, 308–313. [CrossRef]
6. Amato, F.; Marrone, S.; Moscato, V.; Piantadosi, G.; Picariello, A.; Sansone, C. Chatbots Meet eHealth: Automatizing Healthcare. In Proceedings of the WAIAH@ AI* IA, Bari, Italy, 14–17 November 2017; pp. 40–49.
7. Giacalone, M.; Salehi, S. CREA: An introduction to conflict resolution with equitative algorithms. In *Algorithmic Conflict Resolution*; Romeo, F., Dall'Aglio, M., Giacalone, M., Torino, G., Eds.; Editoriale Scientifica: Milan, Italy, 2019.
8. Amato, A.; Amato, F.; Cozzolino, G.; Giacalone, M. Equitative Algorithms for Legal Conflict Resolution. In *Proceedings of the Advances on P2P, Parallel, Grid, Cloud and Internet Computing: Proceedings of the 14th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2019) 14, Antwerp, Belgium, 7–9 November 2019*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 589–597.
9. Amato, F.; Mazzeo, A.; Moscato, V.; Picariello, A. A system for semantic retrieval and long-term preservation of multimedia documents in the e-government domain. *Int. J. Web Grid Serv.* **2009**, *5*, 323–338. [CrossRef]
10. Amato, A.; Giacalone, M. Data Management in the European Project SCAN. In *Proceedings of the Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 34th International Conference on Advanced Information Networking and Applications (WAINA-2020)*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 978–983.
11. Amato, A.; Giacalone, M. Analysis of Data for SCAN Project. In *Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 34th International Conference on Advanced Information Networking and Applications (WAINA-2020), Caserta, Italy, 27–29 March* 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 970–977.
12. Caldarini, G.; Jaf, S.; McGarry, K. A Literature Survey of Recent Advances in Chatbots. *Information* **2022**, *13*, 41.

13. Mauldin, M. ChatterBots, TinyMuds, and the Turing Test: Entering the Loebner Prize Competition. In Proceedings of the Twelfth AAAI National Conference on Artificial Intelligence, Seattle, WA, USA, 31 July–4 August 1994
14. Abdul-Kaer, S.A.; Woods, J. Survey on chatbot design techniques in speech conversation systems. *Int. J. Adv. Comput. Sci. Appl.* **2015**, *6*. [CrossRef]
15. Jonell, P. Using Social and Physiological Signals for User Adaptation in Conversational Agents. In Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS, Montreal, QC, Canada, 13–17 May 2019; Volume 4, pp. 2420–2422.
16. Chen, H.; Liu, X.; Yin, D.; Tang, J. A Survey on Dialogue Systems: Recent Advances and New Frontiers. *ACM SIGKDD Explor. Newsl.* **2018**. [CrossRef]
17. Luo, B.; Lau, R.Y.K.; Li, C.; Si, Y. A critical review of state-of-the-art chatbot designs and applications. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2022**, *12*, e1434. [CrossRef]
18. Weizenbaum, J. ELIZA—A computer program for the study of natural language communication between man and machine. *Commun. ACM* **1966**, *9*, 36–45. [CrossRef]
19. Wallace, R. *The Elements of AIML Style*; ALICE A. I. Foundation, Inc.: San Francisco, CA, USA, 2003.
20. Campbell, M. Prizewinning chatbot steers the conversation. *New Scientist*, 26 October 2010.
21. Pudner, K.; Crockett, K.; Ieee, M.; Bandar, Z. An Intelligent Conversational Agent Approach to Extracting Queries from Natural Language. In Proceedings of the World Congress on Engineering 2007 Vol IWCE 2007, London, UK, 2–4 July 2007.
22. Nazir, A.; Yaseen, M.; Ahmed, T.; Imran, S.; Wasi, S. A Novel Approach for Ontology-Driven Information Retrieving Chatbot for Fashion Brands. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 546–552. [CrossRef]
23. Wanner, L.; André, E.; Blat, J.; Dasiopoulou, S.; Farrùs, M.; Fraga, T.; Kamateri, E.; Lingenfelser, F.; Llorach, G.; Martinez, O.; et al. KRISTINA: A Knowledge-Based Virtual Conversation Agent. In Proceedings of the International Conference on Practical Applications of Agents and Multi-Agent Systems, Porto, Portugal, 21–23 June 2017; pp. 284–295.
24. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119. [CrossRef]
25. Franco, M.F.; Rodrigues, B.; Scheid, E.J.; Jacobs, A.; Killer, C.; Granville, L.Z.; Stiller, B. SecBot: A business-driven conversational agent for cybersecurity planning and management. In Proceedings of the 2020 16th International Conference on Network and Service Management (CNSM), Izmir, Turkey, 2–6 November 2020; IEEE: Piscataway, NJ, USA; pp. 1–7.
26. Bocklisch, T.; Faulkner, J.; Pawlowski, N.; Nichol, A. Rasa: Open Source Language Understanding and Dialogue Management. *arXiv* **2017**, arXiv:1712.05181. [CrossRef]
27. Boonstra, L. *Definitive Guide to Conversational AI with Dialogflow and Google Cloud*; Springer: Berlin/Heidelberg, Germany, 2021.
28. Mitrevski, M.; Mitrevski, M. Getting started with wit. ai. *Developing Conversational Interfaces for iOS: Add Responsive Voice Control to Your Apps*; Apress: New York, NY, USA, 2018; pp. 143–164.
29. Serban, I.V.; Sankar, C.; Germain, M.; Zhang, S.; Lin, Z.; Subramanian, S.; Kim, T.; Pieper, M.; Chandar, S.; Ke, N.R.; et al. A Deep Reinforcement Learning Chatbot. *arXiv* **2017**, arXiv:1709.02349. [CrossRef]
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**. [CrossRef]
31. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training; OpenAI: San Francisco, CA, USA, 2020.
32. Lock, S. What is AI chatbot phenomenon ChatGPT and could it replace humans? *The Guardian*, 5 December 2022.
33. Amato, A.; Giacalone, M. Quality Control in the Process of Data Extraction. In *Web, Artificial Intelligence and Network Applications: Proceedings Workshops of the 34th International Conference on Advanced Information Networking and Applications (WAINA-2020), Caserta, Italy, 15–17 April 2020*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 993–1002.
34. Chen, Q.; Zhuo, Z.; Wang, W. Bert for joint intent classification and slot filling. *arXiv* **2019**, arXiv:1902.10909.
35. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.* **2020**, *53*, 1–34. [CrossRef]
36. Suneera, C.; Prakash, J. A bert-based question representation for improved question retrieval in community question answering systems. In Proceedings of the Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019; Springer: Berlin/Heidelberg, Germany, 2021; pp. 341–348.
37. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
38. Chandra, A.; Stefanus, R. Experiments on paraphrase identification using quora question pairs dataset. *arXiv* **2020**, arXiv:2006.02648.
39. Peyton, K.; Unnikrishnan, S. A comparison of chatbot platforms with the state-of-the-art sentence BERT for answering online student FAQs. *Results Eng.* **2023**, *17*, 100856. [CrossRef]
40. Damodaran, P. Parrot: Paraphrase Generation for NLU. 2021. Available online: https://github.com/PrithivirajDamodaran/Parrot_Paraphraser (accessed on 25 March 2023).
41. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.

42. Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T.Y. Mpnet: Masked and permuted pre-training for language understanding. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 16857–16867.
43. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
44. Hermans, A.; Beyer, L.; Leibe, B. In defense of the triplet loss for person re-identification. *arXiv* **2017**, arXiv:1703.07737.
45. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. Llama: Open and efficient foundation language models. *arXiv* **2023**, arXiv:2302.13971.
46. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. React: Synergizing reasoning and acting in language models. *arXiv* **2022**, arXiv:2210.03629.