

Article

Neural Embeddings for the Elicitation of Jurisprudence Principles: The Case of Arabic Legal Texts

Nafla Alrumayyan *  and Maha Al-Yahya 

Information Technology Department, College of Computer and Information Sciences, King Saud University, P.O. Box 145111, Riyadh 4545, Saudi Arabia; malyahya@ksu.edu.sa

* Correspondence: naflaru@gmail.com

Abstract: In the domain of law and legal systems, jurisprudence principles (JPs) are considered major sources of legislative reasoning by jurisprudence scholars. Generally accepted JPs are often used to support the reasoning for a given jurisprudence case (JC). Although eliciting the JPs associated with a specific JC is a central task of legislative reasoning, it is complex and requires expertise, knowledge of the domain, and significant and lengthy human exertion by jurisprudence scholars. This study aimed to leverage advances in language modeling to support the task of JP elicitation. We investigated neural embeddings—specifically, doc2vec architectures—as a representation model for the task of JP elicitation using Arabic legal texts. Four experiments were conducted to evaluate three different architectures for document embedding models for the JP elicitation task. In addition, we explored an approach that integrates task-oriented word embeddings (ToWE) with document embeddings (paragraph vectors). The results of the experiments showed that using neural embeddings for the JP elicitation task is a promising approach. The paragraph vector distributed bag-of-words (PV-DBOW) architecture produced the best results for this task. To evaluate how well the ToWE model performed for the JP elicitation task, a graded relevance ranking measure, discounted cumulative gain (*DCG*), was used. The model achieved good results with a normalized *DCG* of 0.9 for the majority of the JPs. The findings of this study have significant implications for the understanding of how Arabic legal texts can be modeled and how the semantics of jurisprudence principles can be elicited using neural embeddings.

Keywords: neural embeddings; paragraph vector; document embedding; doc2vec; jurisprudence principles; Arabic legal text; Arabic language; Sharia law



Citation: Alrumayyan, N.; Al-Yahya, M. Neural Embeddings for the Elicitation of Jurisprudence Principles: The Case of Arabic Legal Texts. *Appl. Sci.* **2022**, *12*, 4188. <https://doi.org/10.3390/app12094188>

Academic Editor: Kuei-Hu Chang

Received: 13 March 2022

Accepted: 17 April 2022

Published: 21 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Jurisprudence principles (JPs), or legal maxims, are considered a major source of legislative reasoning. A JP is defined as “an all-inclusive rule based on legal evidence written accurately in comprehensive words and applies to all of its related particulars” [1]. JPs are also defined as theoretical abstractions that are expressed as short and concise statements that are expressive of the goals and objectives of a specific legal rule [2]. JPs are influential in shaping legal decisions and opinions [3] and provide guidance and support for scholars in deducing a ruling for a specific jurisprudence case (JC). Such elicitation is not a simple task. It requires the expertise and knowledge of scholars in the domain and significant and lengthy individual exertion. Identifying the appropriate JP for the given JC involves manual analysis of Arabic legal texts (in the form of Sharia law texts) and a high level of semantic reasoning. This process is lengthy and costly in terms of time and human effort. Moreover, the semantics of the relationship between the JPs and their corresponding JCs are not explicit and cannot be easily realized from word usage in the JC textual description. Thus, the research question we aim to address in this study is as follows: “Given a jurisprudence case, to what extent can neural embedding models provide satisfactory results for the task of eliciting and identifying the most relevant jurisprudence

principle?”. There are numerous JPs used in Arabic legal (Sharia law) texts; however, this study will focus on the five major principles:

- JP 1: Matters are determined according to intentions;
- JP 2: Certainty is not overruled by doubt;
- JP 3: Hardship begets facility;
- JP 4: Harm must be eliminated;
- JP 5: Custom is a basis for judgment.

The remainder of the paper is structured as follows. In Section 2, we provide a brief overview of neural embeddings and the different architectures used for natural language processing (NLP) tasks. Section 3 presents relevant works that use neural embeddings for Arabic NLP tasks. Section 4 describes the methodology, including the corpus development, experiments, and evaluation method. In Section 5 we discuss the results obtained, while in Section 6 we present our conclusions and suggestions for future work.

2. Neural Embeddings

Neural embeddings are representations that enable a wide range of applications in NLP. An embedding uses contextual information to represent a word based on its context. These language models are capable of representing the statistical structure of the written text in ways that can adequately solve many NLP problems. An essential element in the generation of neural embedding models is the vectorization process, which involves applying tokenization and then associating a numeric vector with each token. There are several methods for associating a token with a vector; the two popular methods are one-hot encoding and word embeddings. One-hot encoding consists of associating an integer, index i , with each word, which is converted to a binary vector that is the size of the vocabulary. The vector will have a value of 0 for all elements except the element in position i , which will have a value of 1. Word embeddings [4] are another method for associating a token with a vector; they are learned from data and not hard-coded as in the one-hot encoding method. Word embeddings are low-dimensional dense vectors, unlike one-hot encodings, which are binary, sparse (mostly zeros), and very high dimensional.

There are two methods for obtaining word embeddings for a given task. The first is acquiring the embeddings through the main task (such as classification or retrieval); here, the embeddings are learnt starting from random vectors. The second method is to use a word embedding model that has been previously trained for another task. In this study we use the first approach, since there are no pre-trained word embeddings that we can utilize for Arabic legal texts.

One of the popular methods for the computing of word embeddings from an unlabeled corpus of text is the word2vec algorithm [5]. Word2vec is, therefore, considered an unsupervised learning approach. Instead of providing labels to identify the meanings of target words in the text, the algorithm learns to predict words near the target word in the sentence; the labels, then, are the nearby words. What is important in this model is not the predictions but rather the vector that the algorithm builds to generate those predictions. There are two methods to train a word2vec model, skip-gram and continuous bag-of-words (CBOW). The skip-gram approach predicts the context of the words (output words) from the word of interest (input word), whereas the CBOW predicts the target word (output word) from the nearby words (input words).

More interesting for our purposes are models that go beyond the word level to the document level and provide sentences or phrase-level representations, such as document vector models [6] that extend the word embedding model (word2vec) and sentence vector model (Sentence2vec) [7]. These representations reflect similarities and dissimilarities between words and documents by grouping the vectors of similar words and documents in a low-dimensional vector space. Document embedding is an unsupervised learning technique for representing each document as a vector. The algorithm learns fixed-length feature representations from variable-length pieces of texts such as sentences, paragraphs, and documents [6]. The model is trained to map every paragraph to a unique vector,

and every word is also mapped to a unique vector. There are two architectures for this model, the paragraph vector distributed memory model (PV-DM) and the paragraph vector distributed memory model without word ordering (PV-DBOW).

The PV-DM considers the average of the paragraph vector with each of the word vectors in the paragraph itself to predict the next word in a text window. Each paragraph and word is mapped to a unique vector. The paragraph is represented by a column in matrix D , and the word is represented by a column in matrix W . The paragraph word vectors are averaged, or concatenated, to predict the next word within a context. The inference stage is used to compute the paragraph vector D for a previously unseen new paragraph.

The PV-DBOW ignores the context of the word in the input and treats each document input as a bag-of-words. Rather than learning word vectors, the algorithm learns the vectors of paragraphs and is trained to predict the words in the document. This model does not have to store as much data; it only stores the softmax weight without word vectors. To predict words, the model randomly samples from a paragraph at each iteration of the stochastic gradient descent. Given the paragraph vector, it samples a text window and then a random word from the text window and conducts a classification task.

An embedding model that has shown encouraging results for text classification is the task-oriented word embedding model (ToWE) [8]. This model applies the learning embeddings of the words based on the semantic or task-specific features of the corpus. The study presented in [8] indicates that using the Word2Vec Genism implementation for the word embedding would not improve the performance in the text classification task because the words' functional features in real tasks are ignored in the training process; ToWE, designed for text classification, solves that problem.

3. Related Work

In this section we review studies that use neural embeddings as a language representation model for tasks in Arabic NLP. A growing number of scholars are employing word or document embedding as a representational model for information retrieval (IR) and NLP-related tasks. This approach has been shown to significantly improve performance in many NLP tasks. El Mahdaouy et al. [9] introduced the semantically enhanced term frequency (SMTF) approach, which is a modified method used to compute term frequency. Representations of words are distributed in a vector space (word embedding) based on semantic similarities. This approach seeks to enhance the relevance scores that enable distinct but semantically related terms to be matched with one another. Semantically similar terms are integrated and weighted without modifying the document or query growth functions. Document scoring was improved and mismatches caused by the bag-of-words assumptions were decreased by enhancing the document term frequency among the target query term and its semantically similar word. The retrieval process for documents written in Arabic depends on the bag-of-words paradigm, which is based on semantic similarities between word vectors. The SMTF was evaluated at the inText REtrieval Conference (TREC), and the results demonstrated that incorporating the enhanced term frequency in standard probabilistic IR models produced significant improvements in comparison with baseline bag-of-word models.

Researchers expanded the scope of this initial work by comparing SMTF-based extensions with word-embedding-based IR models. For example, El Mahdaouy et al. [10] developed an approach to deal with term mismatches in IR by incorporating words embedding semantic similarities into existing IR models. They achieved this by following an axiomatic approach to semantic term matching. Their work was an extension of existing IR frameworks, including the BM25, as well as information-based and language models. The effectiveness of these models depends on the integration of word similarity into the scoring functions of previous models. As part of the same study, El Mahdaouy et al. also conducted several experiments on the standard Arabic TREC data set, using title description fields on the Terrier 3.5 IR platform. They investigated three neural word embedding

models: the CBOW, the skip-gram model (SKIP-G), and the GloVe. The SKIP-G and GloVe produced results that were similar and significantly better than the baseline CBOW. The models significantly outperformed both the semantic indexing approach and the three word-embedding-based IR models (LM+WE, NTLM, and GLM).

Zahran et al. [11] investigated different vectorized space representation techniques in Arabic. The authors used CBOW, SKIP-G, and GloVe to construct word representations in a vector space. Each of the models was programmed to contain a wide range of Arabic text sources. Following the normalization process, the three models were aligned to the combined corpus. Intrinsic and extrinsic evaluations were then conducted to assess the impacts on two NLP applications: IR and short-answer grading. The neural word embedding models, including word2vec and GloVe, outperformed the previous query around expansion and enhanced the retrieval process more than other semantic expansion techniques.

Furthermore, these models, compared with previous models, have performed better on information retrieval tasks. Ganguly et al. [12] developed a technique that used a generalized language model (GLM) as part of a word-embedding-based information retrieval language model. They used vectors to embed words and model term dependencies to derive the transformation probabilities between words. Using the GLM, a generative process transformed a term, t , into a term t observed in the query. This work was based on the assumption that words in a query could be generated independently in one of three ways: LM baseline, transformation via document sampling, or transformation via collection sampling. The three methods were combined in the scoring function to formulate the final GLM. Several experiments were conducted with TREC data sets, and the results generated outperformed those produced by the baseline LM and LDA-LM IR models.

A handful of authors have examined the role of word embedding within a retrieval model, in particular the translation language model. Zuccon et al. [13] proposed the adoption of a neural translation language model (NTLM) for IR. Their research focused on neural word embedding models, such as CBOW and SKIP-G, and their use within a translation language model. Translation probabilities were employed to estimate the application of word embedding. More specifically, cosine similarity was used to obtain the probability distribution of the translation. These approaches were then evaluated against the translation language model based on mutual information (TLM-MI) and Dirichlet smoothing baseline (Dirichlet LM). The subsequent analysis was conducted using four standard TREC collections for informal retrieval. The results showed a marked improvement in the translation language model. Indeed, the NTLM approach achieved better performance in terms of IR than the baseline language model and other state-of-the-art translation language frameworks.

Various methods that use word embedding vectors have been proposed to improve the performance of language models in IR. Vulić and Moens [14], for example, introduced a bilingual word embedding skip-gram (BWESG). This unified technique established a language modeling framework that could be employed in IR. MoIR was identified for use in the analysis of monolingual data retrieval and CLIR was identified for use in cross-lingual information retrieval. This framework relied on the estimation of bilingual embedding vectors through the use of single-word embedding, based on word occurrences in the target document. It was also reliant on the vocabulary size of the text as a compositional approach. Several experiments were conducted on English and Dutch language samples, using benchmarking CLEF 2001–2003 collections. The authors reported significant improvements in performing both monolingual IR and cross-lingual IR tasks by adopting a linear combination of document embedding and a baseline language model. The proposed models were superior to LDA-based IR models.

Recent research on word embedding models has clarified that most existing word representation methods only consider contextual information, which is suboptimal when they are used in special tasks because of the lack of task-specific features. Liu et al. [8] proposed a task-oriented word embedding method that can capture both the semantic

features and task-specific features of words, especially in text classification tasks. This method incorporates task-specific features into the training process to reveal the words' functional attributes in the embedding space. They released the source code for this novel version of word2vec, and they investigated its empirical performance on five text classification data sets. The proposed methods significantly outperformed the state-of-the-art methods.

Numerous studies have sought to apply neural embeddings approaches and techniques in machine learning applications. For example, Lee et al. [15] utilized a document vector to generate a labelled data set by using an unsupervised learning approach on a small portion of a labelled data set. They proposed the concatenate vector as a representation of the document vector. The concatenate vector is a combination of the document and polarity vectors. Every document in the training data set was tagged as positive or negative. Then, the doc2vec algorithm was trained to have only two vectors, a positive documents vector and a negative document vector. Inferences were drawn from the training and testing data with this doc2vec to generate polarity. The resulting vector was used to train a support vector machine (SVM) classifier to test the documents' sentiments.

Agrawal et al. [16] used document embedding to learn vector representations for a paragraph or sentence and develop a process for classifying tweets that spread misinformation. The doc2vec model was trained on a sentiment corpus that consisted of 1.5 M tweets. The model was used to obtain vector representations for the tweets in the testing data sets. Preliminary classification experiments showed that the doc2vec model yielded better results than n-gram-based features.

Recently, several studies have used document or word vectors as feature extraction systems and used the vectors to design different distance metrics for question-and-answer data sets. Belinkov et al. [17] used doc2vec and word2vec to compute the vectors of a question-answer data set. The vector similarity features were used as input for a multi-class linear SVM classifier. The approach was evaluated in both English and Arabic and was effective in both. Similarly, Tran et al. [18] used a word vector and topic model to train an SVM classifier for answer quality prediction.

Douzi et al. [19] proposed a new spam-filter-based document embedding approach (a paragraph vector) to capture embedded information in email messages, relevant features of that information, and similar content and features in other emails. Two vectors were used to represent each email: one representing the message's contexts and other the message's features. The correct label of the email, together with these two vectors, was used to train a logistic regression classifier. The experiment proved that the proposed spam filter overcame the limitation of the bag-of-words approach because it captured the relationship that exists between words and their context.

In comparison with other languages, we observed a lack of effort related to Arabic NLP research in the legal domain. The complexity of the Arabic language may be a cause, as may be the lack of supporting resources, such as adequate annotated corpora, reliable related tools, and algorithms. Regarding Arabic legal texts, some research efforts have been conducted for a number of NLP tasks, such as semantic annotation of legal texts, Arabic legal ontology, and legal text classification.

For semantic annotation of legal texts, several approaches have been proposed to automatically classify the texts in the legal domain based on the semantic annotation process. Research has proposed a model for annotating normative provisions in Arabic legal texts [20]; a rule-based scheme has been utilized to identify explicit linguistic markers commonly used by legislators to identify specific provision categories in Arabic legal texts. Another study was introduced in [21], where the authors presented an automated tool to semantically annotate domain-specific Arabic web documents with the help of ontologies. The objective was to incorporate semantic notions in the web searching process to enhance the overall process and the retrieval accuracy. Another research study [22] proposed an approach to recognize and identify the semantic structure of Arabic legal documents.

This structural information would help with document annotation, forming a useful and coherent infrastructure ready for IR.

In the area of Arabic legal ontologies, the development of legal ontologies has increased significantly with the diversity of approaches and their application to Arabic text. The formalization of law through legal ontologies has proved the increasingly prominent role of ontologies in representing, processing, and retrieving legal information. A research study introduced the Arabic Legal Query Expansion System (ALES) for information retrieval for Arabic legal texts [23]. The proposed model empowered by an Arabic domain ontology assists in extracting legal terms from related documents. Moreover, another study on legal information retrieval proposed an approach for ontological learning using Tunisian legal texts [24]. Their suggested search process exploits the user's profile and uses a query reformulation mechanism based on the learned ontology.

Other research was based on the construction of an ontology application for the legal domain. [25]. This work was based on the semantic content of jurisprudence decisions being deployed at several system levels to research decisions. A recent research study proposed a middle-out approach that allows the rapid construction of a well-founded ontology [26]. The ontology aims to support users in describing a specific legal situation and retrieving relevant legal articles and court decisions in similar Moroccan commercial law cases. Bottom-up construction is another ontological approach proposed for building a legal-domain-specific ontology from Arabic texts. In their work [27], the authors used a bottom-up approach that utilized the Arabic WordNet (AWN) project to enrich the ontological vocabulary. Another approach is to exploit the document properties to generate ontological relationships, such as the work presented in [28], which investigates the semi-automatic learning of an ontology from Arabic legal documents.

Regarding legal text classification, one research study [29] examined text classification techniques for Arabic legal texts. They proposed a system that automatically detects the judgment class to support law professionals. The study presented four classification models used to classify Arabic legal documents: naïve Bayes, support vector machine, decision tree, and k-nearest neighbor. The system was designed specifically for Moroccan court legal documents. In the area of legal argumentation, a recent study [30] focused on the field of argumentation mining at the level of Arabic legal texts. They proposed a framework to extract and detect the components of arguments in Arabic legal document text. A supervised learning approach that integrates an annotated Arabic legal text corpus was used to identify the components of arguments from legal decision texts as a final goal.

In a review of the literature, we did not find any studies that used neural embedding language models on Arabic legal texts (Sharia law) and specifically for the elicitation of JPs. The study presented in this article aims to fill this gap and evaluate how well neural embeddings can elicit JPs that are relevant to a specific JC.

4. Methodology

To investigate whether neural embeddings as representations are suitable for identifying and eliciting JPs from JCs, the following methodology was adopted. First, a data set was developed to serve as the corpus for learning the embeddings. Next, several experiments were conducted for training and tuning the model parameters to find the optimal settings for the task. Finally, the resulting models were evaluated, using the ranking performance of the most similar JCs as the measure. We used discounted cumulative gain (*DCG*) as the ranking measure [31]; *DCG* is a ranking quality measure employed in evaluating the effectiveness of web search engine algorithms and similar applications. It uses graded relevance as a measure of usefulness (i.e., the gain from examining a document). This gain is accumulated starting at the top of the ranking and may be reduced or discounted at lower ranks. *DCG* is the total gain accumulated at a particular rank p and is defined as:

$$DCG_p = rel_1 + \sum_{i=2}^p 2^{\frac{rel_i}{\log_2 i}} \quad (1)$$

We conducted our study using the following steps:

- Produce the ideal DCG at position p ($iDCG_p$) to compare the performance on a list sorted by relevance. For our task, the relevance score ranges from 0 to 3:
 - Legal topic match = No, JP Match = No → Score = 0
 - Legal topic match = Yes, JP Match = No → Score = 1
 - Legal topic match = No, JP Match = Yes → Score = 2
 - Legal topic match = Yes, JP Match = Yes → Score = 3
- Compute the normalized version of DCG , with the results between 0 and 1, using this formula:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \quad (2)$$

4.1. Dataset

As there is no available corpus for Arabic legal JCs based on Sharia law, we developed our own JC corpus for the experiments. We called this corpus the JC corpus. We used a key reference text used by legal scholars (Sharia scholars), Almuḡni's Book of JCs [32], as the source for the data. This multivolume series is one of the largest and most widely used texts for JCs. Its subject matter is classified into 68 legal topics, each with 88 sections; it contains 7330 individual JCs. A link to the book which shows the JCs organized into legal topics can be found here [32].

4.2. Preprocessing

Preprocessing involved transforming the collected Arabic documents into texts suitable for vectorization. The preprocessing steps required for the experiments and the system implementation included the normalization and the segmentation of the corpus. The normalization process performed on the data included removing special characters and diacritical marks. The PyArabic Python library was utilized to normalize the text [33]. Segmentation is the process of splitting a text document into smaller documents. This process included separating each JC from the book into an individual text document. Since the JCs are all in one single document, this process is necessary. This includes extracting a single JC into an individual text file. Each text file includes a single JC with appropriate information such as the JC title and number. This single text file is treated as a single document for the model. The JC title and JC number are included in these individual documents.

4.3. Experiments

The Gensim library [34] was used to conduct the experiments. It is a Python library used for topic modeling, document indexing, and similarity retrieval with large corpora, and is also utilized for NLP tasks. It includes implementations for the algorithms needed in these experiments, specifically the doc2vec paragraph embeddings algorithms. The platform that was used to conduct the experiments was Jupyter Notebook 5.3.1, which is a Python-Integrated Development Environment (IDE).

4.3.1. Experiment 1: Using a Supplementary JP Corpus

In the first experiment, we wanted to investigate whether document embeddings are capable of modeling the semantic similarity between a JC and its relevant JP using standard definitions of JP found in legal texts (Sharia texts). To accomplish this, we first built a supplementary corpus for JP definitions collected from two JP books [35,36]. We called this corpus the JP corpus. The corpus consists of five documents, one for each of the five principles (JP1, JP2, JP3, JP4, and JP5). Each document includes definitions, explanations, and sample JCs. The vocabulary size of the corpus is approximately 1.8 million tokens.

Next, the Doc2vec model was trained on the combined corpora (the JC corpus as well as the JP corpus).

According to the literature [6], one of the best performing doc2vec algorithms is the PV-DM. It is usually adequate for NLP tasks such as similarity and retrieval, providing state-of-the-art performance. Therefore, this model was chosen for our first experiment. We used settings for the values of the hyperparameters that were similar to those employed in a study that conducted an empirical evaluation of doc2vec [37].

The algorithm iterates through each document and tokenizes the words to train the model using the available vocabulary. Subsequently, the model generates vectors for the words and vectors for the documents (by combining the word vectors). Then, the similarity scores between each of the five JP vectors and the JCs vectors were computed. The results from this experiment were evaluated to see whether the similarity scores between the JCs and JPs resembled that of a case to its principle.

The top 20 JCs were examined for each of the five JPs. The similarity scores were non-indicative of the relevance of a JC to a JP; moreover, a single JC was associated with more than one JP. Interestingly, the JCs expressed in large documents (approximately 1200 words) overpowered the JCs expressed in smaller documents. They were always considered the most similar to each JP, even if they were not relevant to a given JP. This may have occurred because large documents contain sentences and phrases shared with the JP corpus, such as general words that may be contained in any Arabic legal text.

In this first experiment, using the JP definitions corpus did not yield significant results. The JP corpus we used was not sufficient to elicit semantic similarities between a JC and a JP. Therefore, in subsequent experiments, we excluded the JP corpus.

4.3.2. Experiment 2: Using Paragraph Vector Models

In the second experiment, we compared three different architectures for the doc2vec model on the JC corpus only. We applied three paragraph vector models: (1) the PV-DBOW, (2) the PV-DM (which averages the paragraph and words vectors), and (3) the PV-DM (which concatenates the paragraph and words vectors). The hyper-parameter values were tuned to the optimal setting for the task, as shown in Table 1.

Table 1. Hyperparameter values for the three architectures.

Hyperparameter Name and Description	Values for Each Model		
	PV-DBOW	PV-DM Average	PV-DM Concatenate
dm: If the value is 1 then distributed memory (PV-DM) is used; if the value is 0 then distributed bag-of-words (PV-DBOW) is used	0	1	1
vector_size: Dimensionality of the feature vectors	100	100	100
Negative	5	5	5
min_count: Ignores all words with a total frequency < 5	2	2	2
epochs: Number of iterations (epochs) over the corpus	20	20	20
alpha: The initial learning rate	-	0.05	-
window: The maximum distance between the current and predicted word within a sentence	5	10	5
Sample	0	0	0

The three models were trained using the same corpus. To evaluate the models, a random JC was chosen from the corpus and was used to generate the inference vector. The inference stage was used to compute the paragraph vector for a new paragraph. Next, the most similar JC was identified to test whether the models could identify it. The three models were tested over several iterations with random JCs from the corpus. The results

indicated the need for further experimentation to gain a satisfactory result by changing the hyperparameters.

The results of this experiment indicated that the PV-DBOW was the best model for our task. The PV-DM concatenate performed least well. The PV-DBOW was able to find the input case and the case most similar to it. Accordingly, this model was used for subsequent experiments.

4.3.3. Experiment 3: Using Paragraph-Vector with Negative Sampling

Based on the results of the second experiment, which were encouraging for the PV-DBOW, we used the same model for our third experiment but with new values for the hyperparameters. The results changed significantly when the following parameters were changed, as shown in Table 2: window size, negative sample, dbow_words, and sample. For our task, we needed the model to learn the word embeddings only from those context words that could be indicative of a JP, while excluding domain-relevant words that are common or most frequent in Sharia legal texts (so-called noise words).

Table 2. Hyperparameter Values.

Hyperparameter Name and Description	Setting
dm: Distributed bag-of-words (PV-DBOW) is employed	0
vector_size: Dimensionality of the feature vectors	300
Negative: "Noise words" should be drawn	20
min_count: Ignores all words with a total frequency < 5	5
epochs: Number of iterations (epochs) over the corpus	30
dbow_words: Trains word vectors simultaneously with DBOW document vector training	1
window: The word context	3
sample: The threshold for configuring which higher-frequency words are randomly downsampled	0
hs: Negative sampling will be used	0

Negative sampling is a technique that allowed us to achieve this by having each training sample modify only a small percentage of the weights rather than all of the weights. This was accomplished by setting common or most frequent words as negative (those for which we want the network to deliver an output of 0). We assigned the value to 20 to select the common or frequent words as negative and only update the weights for the positive words that were relevant to the five major JPs.

The sample parameter value was also changed to represent the probability of keeping the word in the vocabulary: smaller values for the sample mean that the most frequent words are less likely to be retained. In our task, the most frequent words were general words that may occur in any Arabic legal text (Sharia text). Thus, we increased this value to ensure that only rare words indicative of the JPs remained. Finally, the PV-DBOW is not concerned with the word ordering and ignores the context words in the input. Thus, the word vectors were activated to be trained simultaneously with DBOW document vector. Table 2 presents the values of the parameters used in this experiment.

To evaluate how well the model was capable of eliciting the JP associated with a specific JC, two JCs were chosen randomly for each JP. Using the model, the inference vector was generated for each of the two JCs and the 30 most similar JCs were analyzed by domain experts to evaluate their relevance to the JP (Figure 1 shows a diagram for the evaluation process).

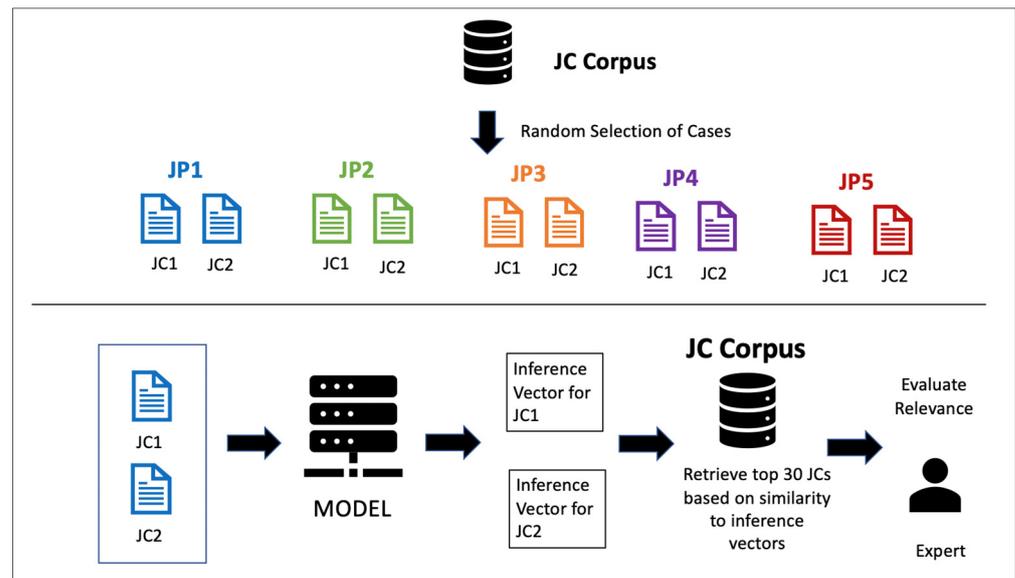


Figure 1. The evaluation process.

The results depicted in Figure 2 show that the model was able to find the JCs most similar to the input JC regardless of the legal topic of the JC. Thus, the similarity produced was based on the shared JP. Interestingly, the model can detect the relevance of JCs based on words that are indicative of a JP.

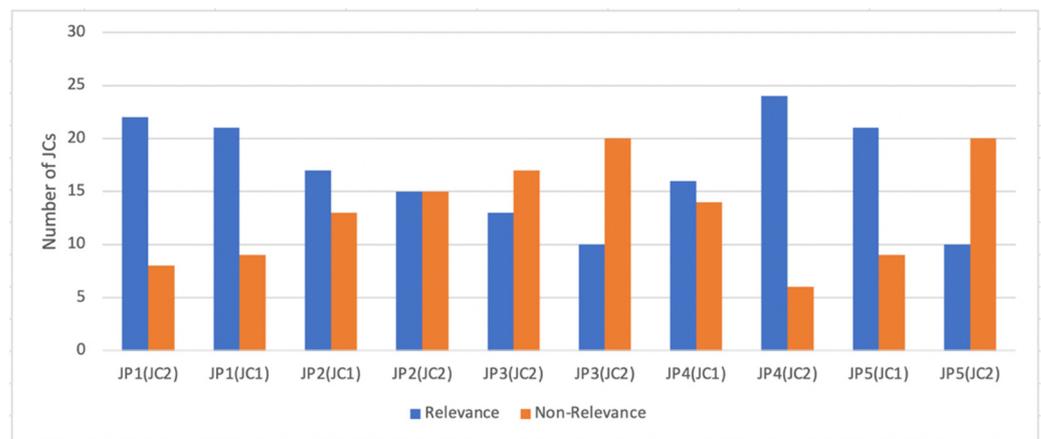


Figure 2. Relevance evaluation of the JCs to JPs.

4.3.4. Experiment 4: Using Paragraph Vectors with Task-Oriented Word Embedding (ToWE)

As the PV-DBOW architecture provided encouraging results, we further investigated the ToWE [8] for the task of JP elicitation. Although the ToWE was originally designed for text classification, we decided to test it for our task because our experiment was similar to a classification task; we considered the five JPs classes, with the objective being to classify a JC with its correct JP class.

The ToWE model is an intelligent learning embedding approach for words based on the semantic and task-specific features of a text. The semantic or context features were specified using the negative sampling introduced in the original word2vec [8]. The task-specific features were dependent on capturing the functional relations between words in the embedding space by using the salient words—those that could be used to distinguish the document category.

Therefore, in the learning framework, if the predicted word (e.g., doubt) is in the salient words' category of one JP (Certainty is not overruled by doubt), the function-aware component will be activated to learn the embedding jointly with the context features. If the word is not salient, then the embedding will be learnt using the original context features only. To model the correlations of JP-salient words, the model constructs a set of word pairs for each salient word. Each word pair contains a positive word and a negative word. The positive words are randomly selected from the words of the target JP or words synonymous to the JP words. The negative words are randomly sampled from the salient words of the other four JPs. The objective function favors higher similarity values for positive word pairs than negative word pairs. Finally, the word vectors of the corpus are generated with awareness of the salient words for each JP and are then exploited to learn document vectors.

To evaluate the ToWE model, five target JCs (relevant to each of the five JPs) were randomly selected. Then, using the model, the most similar JCs were retrieved for each of the five target JCs and reviewed by domain experts to evaluate the relevance of the retrieved JCs to the JP of the target JC (relevant/non-relevant). The results for each JP are presented in Table 3 and depicted in Figure 3.

Table 3. Relevance of results returned for each of the five principles (bold shows best values).

Principle	Percent of Relevant Results
JP1: Matters are determined according to intentions	80%
JP2: Certainty is not overruled by doubt	80%
JP3: Hardship begets facility	90%
JP4: Harm must be eliminated	60%
JP5: Custom is a basis for judgment	90%

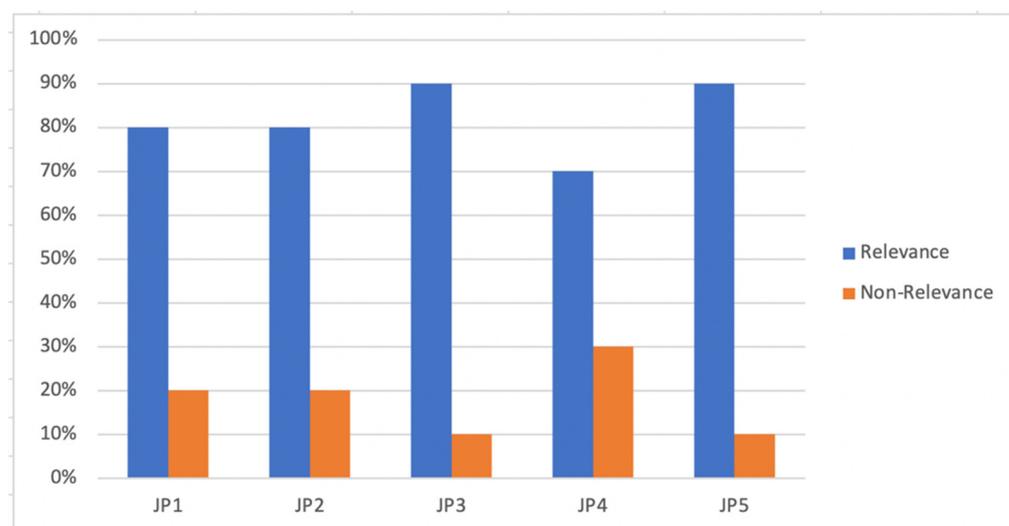


Figure 3. Relevance evaluation of the JCs to JPs for each of the five principles.

5. Results

We present and discuss the results we obtained from the final experiment (Experiment 4) because they are significant. Table 4 shows the ranking performance for each of the five JPs (the ideal *DCG* using the ground truth, and the *DCG* for the JP obtained from the model) along with the average ranking performance (the normalized version of *DCG*, with values ranging from 0 to 1).

Table 4. Ranking performance of each JP.

Jurisprudence Principle	NDCG
JP 1: Matters are determined according to intentions	11.74/12.52 = 0.9
JP 2: Certainty is not overruled by doubt	11.597/12.72 = 0.9
JP 3: Hardship begets facility	12.16/12.16 = 1.0
JP 4: Harm must be eliminated	10.86/12.07 = 0.8
JP 5: Custom is a basis for judgment	13.22/13.46 = 0.9

From the results presented in Table 4, it can be observed that the model performed extremely well for the 50 cases it retrieved based on similarity to the input JC, giving an average normalized DCG value of (0.9) for the majority of the JPs, which is considered a good result [31]. The best performance was achieved with JP 3 (hardship begets facility), with a DCG value of (1.0), and the poorest performance was exhibited by JP 4 (harm must be eliminated), with a DCG value of (0.8).

6. Conclusions and Future Work

In this study we have presented four experiments using neural embeddings for the task of identifying JPs relevant to JCs in the domain of Arabic legal texts (Sharia law). The results of this study are very encouraging. They show that for a new problem, by eliciting the JP relevant to a JC, similarity models built using ToWE with PV-DBOW architecture tuned with the optimal hyper-parameter values can produce satisfactory results. Using neural embeddings, it is possible to model the domain of Arabic legal texts, specifically (Sharia texts) in a way that facilitates the elicitation of the implicit semantics of the text. The semantics reveal the association or relevance of jurisprudence cases to a jurisprudence principle, thereby assisting the process of arriving at a judicial opinion. However, to develop practical solutions, more studies need to be conducted using a larger corpus.

Although the PV-DM model has been shown to be consistently better than PV-DBOW in previous research studies, the results obtained with Arabic legal texts have shown otherwise. The main reason for this is that for our task it is important that the model captures the similarity between the two JCs from different legal topics based on the shared JP.

Although this study provided valuable results and insights, its results were limited by the size of the corpus and the number of labeled JCs available for testing. Moreover, the model presented in this study relies on the corpus of one book only. In future work, other Arabic legal (Sharia law) texts could be included in the corpus for generating the embeddings. In addition, the model only considers the five JPs. There are many other principles associated with JCs; future studies could incorporate those principles as well.

From this study, we can conclude that similarity models built using neural embeddings can produce acceptable results for the task of eliciting JPs. However, more work needs to be done to produce usable and functional solutions.

Author Contributions: Conceptualization, N.A. and M.A.-Y.; methodology, N.A. and M.A.-Y.; software, N.A.; validation, N.A.; formal analysis, N.A. and M.A.-Y.; investigation, N.A. and M.A.-Y.; resources, N.A. and M.A.-Y.; data curation, N.A.; writing—original draft preparation, N.A. and M.A.-Y.; writing—review and editing, N.A. and M.A.-Y.; visualization, N.A. and M.A.-Y.; supervision, M.A.-Y.; project administration, M.A.-Y.; funding acquisition, M.A.-Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a grant from the Researchers Supporting Project (RSP-2021/286), King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This research was supported by a grant from the Researchers Supporting Project No. (RSP-2021/286), King Saud University, Riyadh, Saudi Arabia. The authors would also like to

thank the domain experts who contributed in the human evaluation phase of this study, namely Sulaiman Al-Turki, Luluah AlYahya, Nada Qushami, and Amal Alnafesah.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. *Legal Maxims of Islamic Jurisprudence*; Mishkah: Los Angeles, CA, USA, 2014. Available online: <https://www.muslim-library.com/english/legal-maxims-of-islamic-jurisprudence/> (accessed on 26 October 2019).
2. Kamali, M.H. *Shari'ah Law, An Introduction*; Oneworld: Oxford, UK, 2012. Available online: <https://oneworld-publications.com/shari-ah-law-pb.html> (accessed on 26 October 2019).
3. Saiti, B.; Abdullah, A. The Legal Maxims of Islamic Law (Excluding Five Leading Legal Maxims) and Their Applications in Islamic Finance. *J. King Abdulaziz Univ.-Islamic Econ.* **2016**, *29*, 139–151. [CrossRef]
4. 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; Volume 2, pp. 3111–3119. Available online: <http://dl.acm.org/citation.cfm?id=2999792.2999959> (accessed on 13 December 2018).
5. Mitra, B.; Craswell, N. *An Introduction to Neural Information Retrieval*; Foundations and Trends® in Information Retrieval: Boston, MA, USA, 2017. Available online: <https://www.microsoft.com/en-us/research/publication/introduction-neural-information-retrieval/> (accessed on 17 March 2018).
6. Le, Q.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the 31st International Conference on International Conference on Machine Learning, Beijing, China, 21–26 June 2014; Volume 32, pp. II-1188–II-1196. Available online: <http://dl.acm.org/citation.cfm?id=3044805.3045025> (accessed on 16 March 2018).
7. Pagliardini, M.; Gupta, P.; Jaggi, M. Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 5–6 June 2018; Volume 1, pp. 528–540. [CrossRef]
8. Liu, Q.; Huang, H.; Gao, Y.; Wei, X.; Tian, Y.; Liu, L. Task-oriented Word Embedding for Text Classification. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 2023–2032. Available online: <http://aclweb.org/anthology/C18-1172> (accessed on 9 December 2018).
9. Mahdaouy, A.E.; Alaoui, S.O.E.; Gaussier, É. Semantically enhanced term frequency based on word embeddings for Arabic information retrieval. In Proceedings of the 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), Tangier, Morocco, 24–26 October 2016; pp. 385–389. [CrossRef]
10. Mahdaouy, A.E.; Alaoui, S.O.E.; Gaussier, E. Improving Arabic information retrieval using word embedding similarities. *Int. J. Speech Technol.* **2018**, *21*, 121–136. [CrossRef]
11. Zahran, M.A.; Magooda, A.; Mahgoub, A.Y.; Raafat, H.; Rashwan, M.; Atyia, A. Word Representations in Vector Space and their Applications for Arabic. In *International Conference on Intelligent Text Processing and Computational Linguistics*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 430–443. [CrossRef]
12. Ganguly, D.; Roy, D.; Mitra, M.; Jones, G.J.F. Word Embedding Based Generalized Language Model for Information Retrieval. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 795–798. [CrossRef]
13. Zuccon, G.; Koopman, B.; Bruza, P.; Azzopardi, L. Integrating and Evaluating Neural Word Embeddings in Information Retrieval. In Proceedings of the 20th Australasian Document Computing Symposium, Parramatta, Australia, 8–9 December 2015; pp. 12:1–12:8. [CrossRef]
14. Vulić, I.; Moens, M.-F. Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 363–372. [CrossRef]
15. Lee, S.; Jin, X.; Kim, W. Sentiment Classification for Unlabeled Dataset Using Doc2Vec with JST. In Proceedings of the 18th Annual International Conference on Electronic Commerce: E-Commerce in Smart Connected World, Suwon, Korea, 17–19 August 2016; pp. 28:1–28:5. [CrossRef]
16. Agrawal, T.; Gupta, R.; Narayanan, S. Multimodal detection of fake social media use through a fusion of classification and pairwise ranking systems. In Proceedings of the 2017 25th European Signal Processing Conference (EUSIPCO), Kos Island, Greece, 28 August–2 September 2017; pp. 1045–1049. [CrossRef]
17. Belinkov, Y.; Mohtarami, M.; Cyphers, S.; Glass, J. VectorSLU: A Continuous Word Vector Approach to Answer Selection in Community Question Answering Systems. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, CO, USA, 4–5 June 2015; pp. 282–287. Available online: <http://www.aclweb.org/anthology/S15-2048> (accessed on 29 April 2018).
18. Tran, Q.H.; Tran, V.; Vu, T.; Nguyen, M.; Pham, S.B. JAIST: Combining multiple features for Answer Selection in Community Question Answering. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, CO, USA, 4–5 June 2015; pp. 215–219. Available online: <http://www.aclweb.org/anthology/S15-2038> (accessed on 30 April 2018).

19. Douzi, S.; Amar, M.; Ouahidi, B.E.; Laanaya, H. Towards A new Spam Filter Based on PV-DM (Paragraph Vector-Distributed Memory Approach). *Procedia Comput. Sci.* **2017**, *110*, 486–491. [[CrossRef](#)]
20. Berrazega, I.; Faiz, R.; Bouhafs, A.; Mourad, G. A Semantic Annotation Model for Arabic Legal Texts. In Proceedings of the 9th Hellenic Conference on Artificial Intelligence, Thessaloniki, Greece, 18–20 May 2016. [[CrossRef](#)]
21. Al-Bukhitan, S.; Helmy, T.; Al-Mulhem, M. Semantic Annotation Tool for Annotating Arabic Web Documents. *Procedia Comput. Sci.* **2014**, *32*, 429–436. [[CrossRef](#)]
22. Mezghanni, I.B.; Gargouri, F. Detecting hidden structures from Arabic electronic documents: Application to the legal field. In Proceedings of the 2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA), Towson, MD, USA, 8–10 June 2016; pp. 75–81.
23. Mezghanni, I.B.; Gargouri, F. ALES: An Arabic Legal query Expansion System. In Proceedings of the Conference on Data Science and Knowledge Engineering for Sensing Decision Support (FLINS 2018), Belfast, UK, 21–24 August 2018; pp. 568–575. [[CrossRef](#)]
24. Mezghanni, I.B.; Gargouri, F. Learning of Legal Ontology Supporting the User Queries Satisfaction. In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Warsaw, Poland, 11–14 August 2014; Volume 1, pp. 414–418.
25. Dhoubib, K.; Gargouri, F. Legal application ontology in Arabic. In Proceedings of the Fourth International Conference on Information and Communication Technology and Accessibility (ICTA), Hammamet, Tunisia, 24–26 October 2013; pp. 1–6.
26. Belhoucine, K.; Mourchid, M.; Mouloudi, A.; Mbarki, S. A Middle-out Approach for Building a Legal domain ontology in Arabic. In Proceedings of the 2020 6th IEEE Congress on Information Science and Technology (CiSt), Agadir-Essaouira, Morocco, 5–12 June 2021; pp. 290–295.
27. Belhoucine, K.; Mourchid, M.; Mbarki, S.; Mouloudi, A. A Bottom-Up Approach for Moroccan Legal Ontology Learning from Arabic Texts. In *International Conference on Automatic Processing of Natural-Language Electronic Texts with Noof*; Springer: Cham, Germany, 2020; pp. 230–242.
28. Mezghanni, I.B.; Gargouri, F. Towards an Arabic legal ontology based on documents properties extraction. In Proceedings of the 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), Marrakech, Morocco, 17–20 November 2015; pp. 1–8.
29. Ikram, A.Y.; Chakir, L. Arabic Text Classification in the Legal Domain. In Proceedings of the 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), Marrakech, Morocco, 28–30 October 2019; pp. 1–6.
30. Jasim, K.M.; Sadiq, A.T.; Abdullah, H.S. A Framework for Detection and Identification the Components of Arguments in Arabic Legal Texts. In Proceedings of the 2019 First International Conference of Computer and Applied Sciences (CAS), Baghdad, Iraq, 18–19 December 2019; pp. 67–72.
31. Järvelin, K.; Kekäläinen, J. Cumulated Gain-based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* **2002**, *20*, 422–446. [[CrossRef](#)]
32. المغني لابن قدامة • الموقع الرسمي للمكتبة الشاملة. Available online: <http://shamela.ws/index.php/book/8463> (accessed on 22 December 2018).
33. PyArabic. PyPI. Available online: <https://pypi.org/project/PyArabic/> (accessed on 6 October 2018).
34. Rehurek, R. Gensim: Python Framework for Fast Vector Space Modelling. Available online: <http://radimrehurek.com/gensim> (accessed on 12 December 2018).
35. المتع في القواعد الفقهية. Available online: https://www.goodreads.com/work/best_book/16932078 (accessed on 8 October 2018).
36. الوجيز في إيضاح قواعد الفقه الكلية - المكتبة الوقفية للكتب المصورة. Available online: <http://waqfeya.com/book.php?bid=9501> (accessed on 8 October 2018).
37. Lau, J.H.; Baldwin, T. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, Germany, 11 August 2016; pp. 78–86. [[CrossRef](#)]