PHILIPPINE ENGINEERING JOURNAL
PEJ 2024; Vol. 45, No. 1: 77-94

Received: 24 January 2024
Accepted: 18 June 2024

A Comparative Analysis and Evaluation of Natural Language Processing Document Embedding Techniques on Philippine Supreme Court Case Decisions

Lorenz Timothy Barco Ranera
Mathematical and Computing Sciences Unit, Department of Physical Sciences and Mathematics
College of Arts and Sciences, University of the Philippines Manila
Corresponding Author: lbranera@up.edu.ph

Abstract *– This study explores the application of Natural Language Processing in Philippine law to expedite legal research. It focuses on three document embedding techniques: Doc2Vec, TF-IDF, and OpenAI embedding (text-embedding-ada-002), using a dataset of Philippine Supreme Court Case Decisions from 2015 to 2020 (4,400 case decisions). The objective is to uncover and evaluate semantic relationships between case decisions. Importantly, this paper proposes two evaluation measures, "similarity classification" and "similarity comparison," to evaluate the four embedding models and determine how these captured the semantic similarity relationship between cases. The results show that embedding models performed high accuracy scores in "similarity classification," but performed relatively poorer in the second metric "similarity comparison" with low to moderate accuracy. The best performing model is Doc2Vec with 94% accuracy in "similarity classification" and 72.92% accuracy in "similarity comparison." Future studies can focus on steps to improve performance in "similarity comparison" metric and additional preprocessing techniques such as text reorganization (e.g., summaries, sections). These results clearly demonstrate the potential of document embedding to enhance legal research efficiency in the Philippines and similar domains through Natural Language Processing*.

Keywords*: artificial intelligence, natural language processing, document embedding, legal AI, jurisprudence*

I. INTRODUCTION

The application of Computer Science methodologies, particularly Artificial Intelligence (AI), has been on the rise and is generating excitement, especially within the field of Philippine law [1, 2, 3, 4, 5]. There is a substantial interest and commitment to AI adoption, with certain institutions under the Philippine government incorporating AI to enhance productivity, save time and resources, and deliver higher-quality public services. Moreover, the Philippine government envisions leveraging AI to boost productivity, foster economic growth, and position the nation as a more globally competitive entity [6]. Even the Chief Justice of the Supreme Court of the Philippines, Chief Justice Alexander G. Gesmundo, has consistently voiced his support for the utilization of AI in legal research on three separate occasions. His endorsement is aimed at improving the operations of the Philippine Judiciary [7], expediting the resolution of cases [8], and increasing accessibility to legal references [9]. Since the discipline of law is dominated by language and text [1], Natural Language Processing (NLP), a subfield under AI, enters the scene as a necessary innovation enforced and applied to legal

datasets. It is defined as the set of "computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications" [10]. Document embedding, as one of the techniques under NLP, involves the numerical representation (vectorization) of text data, enabling it to be mapped within an n-dimensional vector space [11]. Several well-known embedding techniques include bag-of-words, n-grams, TF-IDF (Term-Frequency Inverse Document-Frequency), and Doc2Vec, as well as newer methods based on transformers like BERT and OpenAI embedding. Modern embedding techniques, such as Doc2Vec [12, 13], take into consideration the "semantics" between texts. Consequently, the relationships between document vectors also imply semantic similarity relationships between documents [11]. These relationships prove invaluable in tasks such as information/document retrieval, document clustering, and document classification [13, 14].

Regrettably, individuals involved in litigation within the Philippines often encounter the issue of delayed justice. The inefficiencies within the Philippine judiciary can be attributed, in part, to the labor-intensive nature of legal research and various other factors, including an inadequate ratio of courts to the population, lawyers to the population, and slow pace of case disposition. As of May 2020, the population of the Philippines stands at 109,035,343 [15]. With an estimated 2,000 judicial courts across the nation, this translates to just one court for every 50,000 Filipinos [16]. Furthermore, there are approximately 40,000 active Filipino lawyers, resulting in a ratio of just one lawyer for every 2,500 Filipinos [17]. The statistics reveal an average case disposition rate of 0.2545 (the number of cases decided divided by the total number of cases) from 2005 to 2014, with a declining trend observed from 2011 to 2014 [2]. Moreover, the process of legal research among legal professionals in the country is characterized as a labor-intensive and time-consuming endeavor [1]. Chief Justice Gesmundo, a staunch advocate for AI integration in Philippine law, believes that AI has the potential to significantly expedite legal research. He envisions AI as a transformative tool that can make legal searches faster and more accessible, ultimately benefiting the people the judiciary serves [9]. While acknowledging the multifaceted challenges facing the Philippine judiciary, it is imperative to consider technological upgrades and tools that can alleviate certain aspects of the judicial process. For instance, techniques such as document embedding for expedited document retrieval, with its automatic semantic similarity scoring, can offer a valuable solution without the need for exhaustive manual reading.

The primary objective of this paper is to employ three widely recognized document embedding techniques—namely Doc2Vec, TF-IDF, and OpenAI (text-embedding-ada-002)—on a collection of Philippine Supreme Court Case Decisions (Jurisprudence) in order to quantify semantic relationships between any pair of given case decisions. This study considers Supreme Court case decisions from the period spanning 2015 to 2020 with a total of 4,400 case decisions as data set for document embedding. These case decisions serve as the foundational corpus for training four document embedding models: (1) Doc2Vec, (2) TF-IDF, (3) OpenAI using raw data set (hereafter "OpenAI+raw"), and (4) OpenAI using preprocessed data set (hereafter "OpenAI+prepro"). In addition, cosine similarity is used to compute the similarity score between two documents since it is a "widely used" similarity measure [18]. Furthermore, this research attempts to conduct a comparative analysis to determine which among the four embedding models yields the best performance using two proposed evaluation

metrics. However, this study is limited to "true labels" for the two proposed evaluation measures provided by a single legal expert.

These accuracy scores are assessed using two proposed evaluation measures, each designed to accurately capture the essence of "semantic similarity relatedness" among case decisions:

a) Similarity Classification: The first evaluation measure involves classifying two case decisions as either "similar" or "dissimilar" decided based on the cosine similarity score of their document vectors while exploring varying cosine similarity thresholds (posed as a binary classification problem).
b) Similarity Comparison: The second evaluation measure focuses on identifying which of the two case decisions are more similar relative to a third, individual case decision, again, based on the cosine similarity score of their document vectors.

II. RELATED WORKS

2.1 Word Embedding on Philippine Jurisprudence

Within the Philippines, efforts have been made to employ *word embedding* (i.e., word vector representations) and *document embedding* techniques in the realm of jurisprudence. Foreign studies have also delved into domain-specific word embedding models, such as BioWordVec, developed by Zhang, Chen, Yang, Lin, and Lu in 2019 as a contribution to NLP in the field of Biomedical Sciences [19]. Additionally, the Law2Vec embedding model, created by Chaldikis and Kampas in the same year, utilized English legislation from the UK legal system and across the British Commonwealth [20]. These endeavors seemed to pique the interest of three Filipino researchers, Peramo, Cheng, and Cordel II. In 2021, they developed a 300-dimensional word embedding model collectively known as Juris2Vec. This model was built using three prominent word embedding techniques: Word2vec, GloVe, and FastText, applied to Philippine jurisprudence. Their results indicated that Word2vec and FastText word embedding models excelled in "semantic" and "syntactic" evaluation measures, respectively. Through Juris2Vec, the researchers proposed its applicability in classification tasks, information retrieval, word/phrase analogy, translation, and more [3].

2.2 Document Embedding on Philippine Jurisprudence

Document embedding, again, is the process of converting documents from its original form (text encoded in a particular natural human language such as English) to a numerical representation, or a document vector in *n*-dimensions [11]. The application of document embedding model in Philippine legal data set was the objective in the research conducted by Ran era, Solano, and Oco in August 2019. The researchers applied the widely recognized Doc2Vec embedding technique to Philippine case decisions with the aim of expediting legal research for court judges during trials. This approach facilitated quicker judgments by automatically retrieving semantically similar case decisions [2]. The selection of the most effective document embedding technique posed a challenge. However, a study by Mandal, Chaki, and Saha in November 2017 demonstrated that Doc2Vec outperformed three other document embedding techniques: TF-IDF, LDA Topic Modelling, and weighted-averaged Word2Vec, using Indian Jurisprudence [21]. Returning to the study by Ran era, Solano, and Oco, their Doc2Vec model achieved an 80% accuracy rate in similarity classification and

displayed strong monotonicity ($\rho = 0.7691$) when compared with scores provided by a legal expert [2].

2.3 Foreign Applications of Embedding on Legal Corpora

The utilization of document embedding techniques gained considerable attention not only in the Philippines but also in countries with Anglo-American common-law traditions like India. These countries place a strong emphasis on "judicial precedents" (i.e., stare decisis) in determining case outcomes [22, 23]. In a 2011 study, Kumar, P. K. Reddy, V. B. Reddy introduced two distinct methods for identifying similar case decisions in the Supreme Court of India: a text-based approach using TF-IDF and a network-based approach employing bibliographic coupling and co-citation counts. They found that bibliographic coupling and TF-IDF, with specialized preprocessing (removing non-legal tokens), effectively retrieved similar case decisions [24]. Two years later, in 2013, Kumar, P. K. Reddy, V. B. Reddy, and Suri proposed a hybrid approach utilizing TF-IDF embedding and cosine similarity to determine "paragraph links" between different case decision paragraphs. The count of paragraph links served as a measure of similarity, outperforming their previous network-based approach (bibliographic coupling) [25]. However, Mandal, Chaki, and Saha argued in November 2017 that Doc2Vec surpassed the hybrid approach and other embedding techniques, showing the strongest correlation with legal expert scores [21].

There were new pursuits in evaluating embedding techniques in the following years. In 2021, Mandal, Gosh, and Mandal experimented with seven techniques, finding Doc2Vec to perform best based on correlation with a legal expert. TF-IDF, on the other hand, achieved the highest accuracy (87.2%) in binary classification for semantic similarity [26]. In a December 2020 study using a smaller data set of Indian Jurisprudence, Almuslim and Inkpen compared three techniques (Doc2Vec, Word2Vec, and GloVe) and found TF-IDF to be the most effective embedding model according to their metrics [22]. Hence, it's interesting to reconsider TF-IDF as an embedding technique for assessing case decision similarity and explore newer transformer-based models like OpenAI's text-embedding-ada-002. Apart from the Philippines and India, legal researcher Novotna (2020) demonstrated that Doc2Vec has potential in identifying semantically similar Czech jurisprudence, emphasizing the importance of reviewing past decisions in civil-law jurisdictions like the Czech Republic. In her study, she qualitatively evaluated the results of the embedding model with one example (a lease agreement case) where nine out of top ten semantically similar documents shared the same legal issue [27].

III. METHODOLOGY

3.1 Dataset Preparation and Analysis Stage

The data set employed in this study focuses exclusively on Philippine Supreme Court case decisions, known as Jurisprudence, for the years 2015 to 2020. This paper constrained the data set to a specific five-year period due to limitations in time and financial resources, taking into account OpenAI's services operate on a commercial basis (with pricing for every document embedding request). Initially, the initial data set comprised a total of 6,097 cases. The primary language of composition of the cases is English, although there are occasional excerpts in

Filipino and Spanish. The average token size is 7,944.02 tokens, with the largest case containing 421,512 tokens and the smallest comprising 679 tokens, as calculated using the tiktoken Python library.

However, it's important to note that the OpenAI embedding model (text-embedding-ada-002) can handle a maximum of 8,192 tokens in a single API request. Consequently, all case decisions exceeding 8,000 tokens were excluded from the data set [28]. Additionally, case decisions containing fewer than 1,000 tokens were also removed. According to a previous study [2], these shorter documents having a small set of word tokens can mistakenly be classified as "similar" compared to larger documents potentially affecting the performance of the embedding model. As a result, the final data set consists of a total of 4,400 case decisions.

3.2 Preprocessing Stage

The preprocessing stage transforms the saved HTML code of a case decision, obtained from an open-source, free application, into a list of final transformed tokens suitable for embedding modeling. The following preprocessing steps were adopted from related works and implemented using Python. This stage is a prerequisite for three out of the four document embedding models, namely (1) TF-IDF, (2) Doc2Vec, and (3) OpenAI+prepro.

a) The HTML code of a case decision is processed into clean text using the bs4 HTML parser.
b) The clean text is tokenized using wordpunct_tokenize from the nltk Python library. At this stage, the clean text is transformed into a list of initial tokens.
c) All tokens in the initial list are converted to lowercase using the built-in lower function.
d) Tokens that correspond to Filipino names, stopwords, and other insignificant categories (e.g., tokens representing locations, months, adverbs, etc.) are removed from the initial token list.
e) Single-character tokens and tokens containing Unicode and non-alphabetic characters are eliminated from the previous list, as they are likely minor errors or additional unimportant tokens in the case decision.
f) Tokens that appear in fewer than five case decisions or in more than 95% of the case decisions (5,792 out of 6,097 case decisions) are excluded from the previous list. Tokens falling into these extremes of occurrence do not significantly contribute to determining similarity.
g) Each token is lemmatized using WordNetLemmatizer from the nltk Python library to unify different forms of base or stem words (e.g., running, ran, run) into a single token representation (e.g., run).

3.3 Document Embedding Stage

For the document embedding stage, this study attempts to develop four different document embedding models: (1) TF-IDF, (2) Doc2Vec, (3) OpenAI+raw, and (4) OpenAI+prepro.

TF-IDF and Doc2Vec were selected because they were previously applied in judicial case datasets before in past studies [2, 21]. Due to the gaining popularity of generative models [31], OpenAI's embedding model (text-embedding-ada-002) is also selected for comparison and further evaluation. The output of this step is to produce four distinct document vectors from

the four models for each and every 4,400 case decisions (4,400 document vectors for each model and 17,600 document vectors in grand total).

3.3.1 TF-IDF (Term-Frequency Inverse Document-Frequency)

TF-IDF, which stands for "term frequency–inverse document frequency," is one of the most common and early NLP techniques in document vector representation utilized in information retrieval systems during the 2000s but already conceptualized even before [29, 30]. This technique is known for determining unique and special word tokens in a document having higher tf-idf scores as opposed to very common words having lower tf-idf scores [30]. The tf-idf score of a word *w* of a particular document d is computed using the formula (see Equation 1) where $f_{w,d}$ is the number of times *w* appeared in document *d* only, $|D|$ is the total number of documents in a corpus, and $f_{w,D}$ is the number of documents that mentioned word *w* [30].

$$tfidf(w) = f_{w,d} \cdot \log\left(\frac{|D|}{f_{w,D}}\right) \tag{1}$$

In this present study, the TF-IDF embedding model is created and prepared using the TfidfVectorizer from sklearn Python library with the preprocessed data set again as the training data set. The vector size is set to 100. The TF-IDF document vector of every Philippine case decisions is saved in a JSON file stored in a file directory titled ./tfidf_model.

3.3.2 Doc2Vec

Compared to TF-IDF, Doc2Vec is a relatively new document embedding technique conceptualized in 2014 which was initially referred to as "Paragraph Vectors" by its creators from Google company [12, 13]. This technique is known for its ability to capture semantics relationship between documents such that when visualized the vector points of semantically similar documents are proximate to each other in space [13]. The document vectors are produced based from a neural network model of where the probability of a target word (output layer) is computed given the context words and the document where it belongs (input layer). In trying to accomplish (maximize) the high probability of predicting this target word in many iterations, the weights of the document vectors as well as the weights of the context word vectors are repeatedly adjusted from time to time until it reaches their final vector form. In this particular example from the figure below from [12], the context words "*the*", "*cat*", "*sat*," and paragraph *D* are trying to maximize the probability of predicting "*on*" as the next word in the neural network. This is referred to as the Distributed Memory Model of Paragraph Vectors (PV-DM), which is the default architecture in implementing Doc2Vec [12].

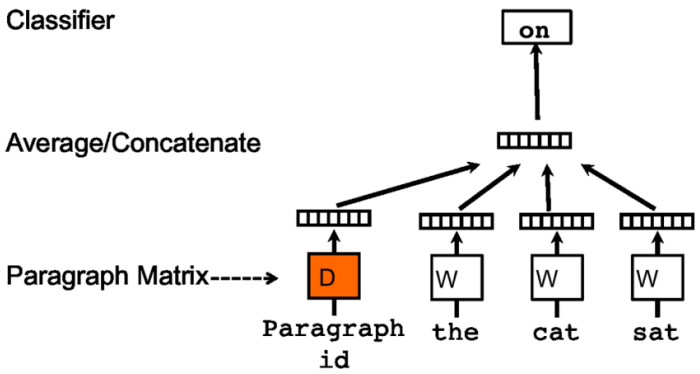

Figure 1. Distributed Memory Model of Paragraph Vectors (PV-DM) [12]

The Doc2Vec embedding model is developed using the Doc2Vec class from gensim Python library with the preprocessed data set as the training data set. The multiprocessing library is also utilized to speedup the training process. For this model, the vector size is 100 and number of epochs is 10 while the other parameters are set in default. The Doc2Vec model file is used to query the Doc2Vec document vector of a particular case decision and retrieve the top-n most similar case decisions.

3.3.3 OpenAI Embedding Model (text-embedding-ada-002)

Due to the popularity rise of generative AI in the past two years [31], this study cannot ignore to include it in its comparative analysis represented by OpenAI, the company responsible for the creation of ChatGPT platform for public consumption powered by their large language model, namely GPT-3.5 and GPT-4. As of writing, their state-of-the-art embedding model is called "text-embedding-ada-002" that is already trained and accessible through API endpoint to convert a text document into a document vector on a commercial basis [28]. Unfortunately, the actual architecture of OpenAI's text-embedding-ada-002 remains confidential outside of the company probably to avoid duplication from competitors [32].

However, their earlier embedding model (text-similarity-davinci-001) was built using a Transformer architecture model [32, 33] according to a published manuscript of the OpenAI embedding model's creators in *arXiv* [34]. The Transformer is also a neural network based model developed in 2017 by machine learning researchers in Google [35]. It is now considered as the "dominant architecture in natural language processing" exceeding established deep-learning techniques such as convolutional neural networks (CNN) and recurrent neural networks (RNN) [36].

3.3.3.1. OpenAI+raw

Compared to TF-IDF and Doc2Vec, the vector size is significantly higher, fixed at 1,536 dimensions. The OpenAI embedding model does not require any preprocessing technique to generate document vectors which saves time and resources demanded during the preprocessing stage as it is traditionally done in Doc2Vec and TF-IDF embedding process. The OpenAI document vectors of the raw text version of Philippine Supreme Court case decisions are obtained through a series of API requests and every response containing the document vectors are saved in JSON file in a file directory named ./openai_model.

3.3.3.2 OpenAI+prepro

This paper is interested to investigate the putative impact of an additional preprocessing step before feeding the data set into a transformer-based embedding model. Another set of document vectors were obtained through OpenAI API requests but using a concatenated string of preprocessed tokens instead of the raw text version. These are saved in the file directory titled ./openai_model_preprocessed.

3.3.4 Cosine similarity

Cosine similarity is used as the only similarity measure in determining the similarity score between two document vectors. The cosine_similarity function from sklearn Python library is utilized for this purpose. However, the threshold (*t*) for determining the metric boundary between "similar" and "dissimilar" is arbitrarily chosen, for example, in [2, 26] the threshold is arbitrarily set at 0.50 while other threshold values were considered in other works. In this study, this study explores varying cosine similarity threshold values ($t \in [0.50, 1.00]$) in order to determine the maximum accuracy scores that can be produced considering Doc2Vec, TF-IDF, and OpenAI document vectors. The formula for the cosine similarity between document vectors *A* and *B* is shown below in Equation 2.

$$sim(A,B) = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2}\sqrt{\sum_{i=1}^{n} {B_i}^2}} \quad (2)$$

IV. RESULTS AND DISCUSSION

To achieve the highest level of semantic similarity relatedness, a document embedding model (the model generating documents vectors) paired with cosine similarity must attain high accuracy scores on two critical evaluation metrics which this study considers and proposes: (1) similarity classification and (2) similarity comparison. These evaluation metrics are considered and proposed because they aim to measure the document embedding model's twin ability: (1) to distinguish between "similar" and "dissimilar" pair of case decisions and (2) to capture the sense of "similarity ordering," thus identifying the "more similar" case between two similar cases, respectively.

4.1 Overview of Test Cases

Fifty (50) tests cases provided for the first evaluation metric in Section 3.2 and forty-eight (48) tests cases for the second evaluation metric in Section 3.3. The number of tests cases is deemed a sufficient number since previous studies evaluated their models using fewer tests cases, namely twenty-five (25) test cases in [2] and forty-seven (47) test cases in [21, 26]. Furthermore, the selection and utilization of tests cases from the training legal dataset (4,400 case decisions) should not cause any issue leading to biased evaluation in the context of this study because document embedding is classified as an unsupervised learning technique in machine learning where "true labels" are not provided. The document embedding models (Doc2Vec, TF-IDF, and OpenAI's text-embedding-ada-002) attempt to reconstruct the similarity relationships among documents in a vector space without the benefit of knowing

which documents are actually similar. The document embedding models merely generates a document vector based on its statistical design and formula. It is only beyond embedding process that it can be identified if a pair of cases are "similar" and "not similar" by checking at the calculated value of the cosine similarity scores and comparing it based on a similarity threshold.

4.2 Performance of Evaluation Metric 1: Similarity Classification

For the first evaluation metric "similarity classification" in Section 3.2., this study prepared randomly selected fifty (50) pairs of case decisions which was manually labeled by a legal expert whether the pair is "similar" and "not similar" as shown in Appendix 1. The balance in the labeled testing data set is ensured comprising of 25 manually labeled "similar" pairs and 25 manually "not similar" pairs. The determination of similarity is obviously subjective, but the author did not prescribe any rubric to provide liberty to the legal expert to decide what is similar or not according to the context of the legal discipline. It was only after labeling task that the legal expert was inquired about his decision-making in labeling the pairs.

According to the legal expert, the order of priority in determining similarity between jurisprudence is the following: (1) legal doctrine/s, (2) legal issue/s, and (3) fact/s of the case. Two cases would be "similar" at least if they share the same legal issue/s, and even "more similar" if they share the same doctrine/s. For example, two criminal cases about the use of illegal of drugs against a special penal law Republic Act 9165 would be "similar" (e.g., pair 3 in Appendix 1) for him, but when a murder case and a drug-related case are compared, then they are "not similar" because these are two distinct crimes (e.g., pair 4 in Appendix 1).

In this evaluation metric, the task is posed as a binary classification problem. In this scenario, the domain expert's manual assessment serve as the "target values" (i.e., "similar/not similar" labels) while the cosine similarity score and similarity threshold value are used to determine the model's prediction ("similar/not similar"). Plotting the threshold (x-axis, $x \in [0.50,0.95]$) against accuracy (y-axis, $y \in [0.00,1.00]$) for the four embedding models, there is a decrease in accuracy for both Doc2Vec and TF-IDF embedding models as the similarity threshold increases from 0.50 to 0.95 albeit a slower decrease in TF-IDF as opposed to Doc2Vec. Furthermore, the maximum of their respective accuracy of TF-IDF and Doc2Vec are located around the threshold value of 0.50 (see Figures 2 and 3). Conversely, the accuracy of the two OpenAI embedding models, namely OpenAI+raw and OpenAI+prepro, experiences a significant increase as the similarity threshold approaches the range of 0.8 to 0.9 (see Figures 4 and 5). The most optimal score threshold is summarized in Table 1 based on the maximum accuracy produced.

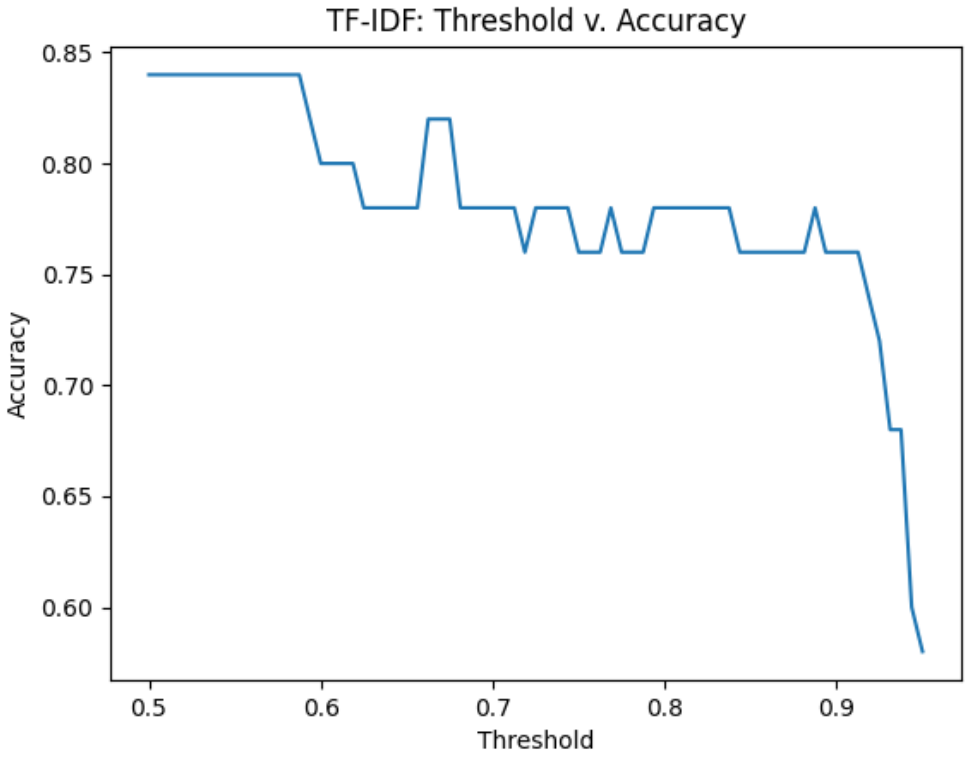

Figure 2. TF-IDF: Threshold-Accuracy Plot

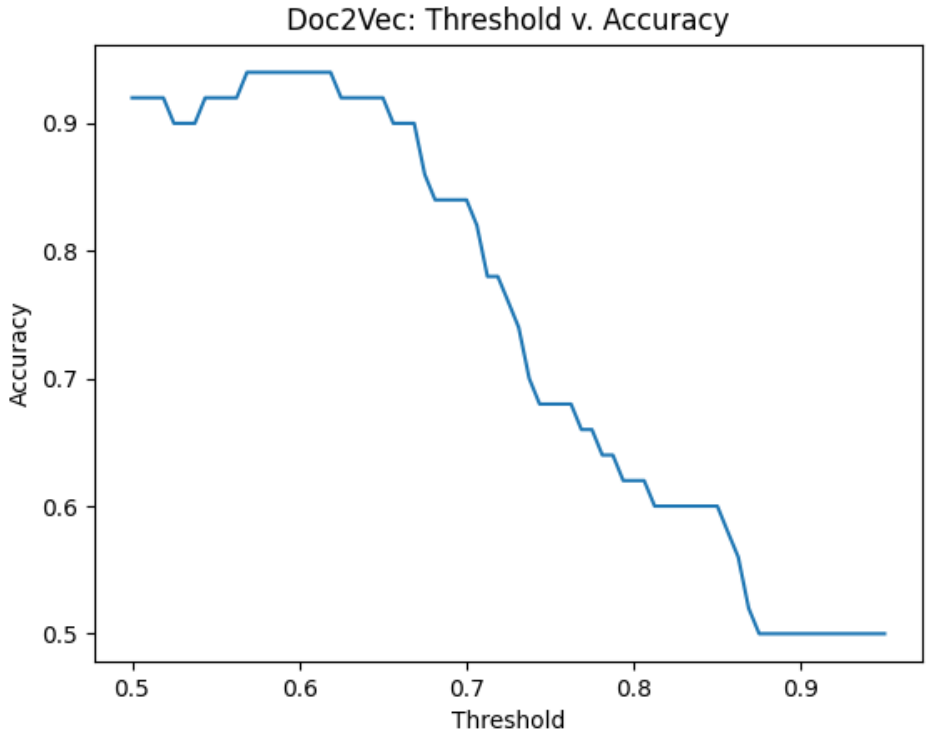

Figure 3. Doc2Vec: Threshold-Accuracy Plot

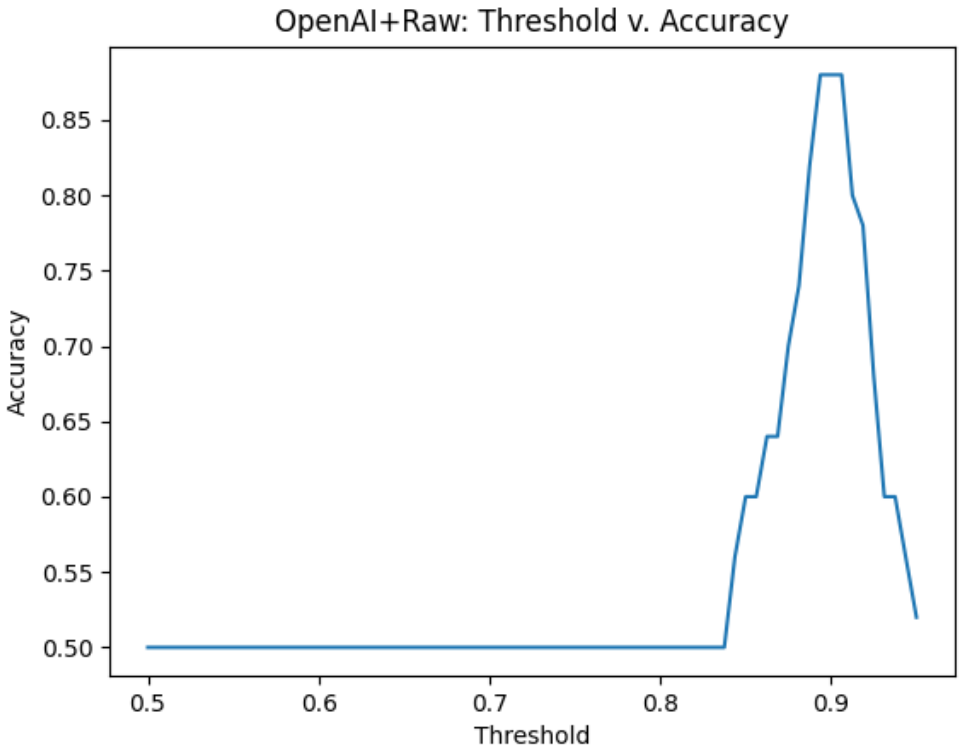

Figure 4. OpenAI+raw: Threshold-Accuracy Plot

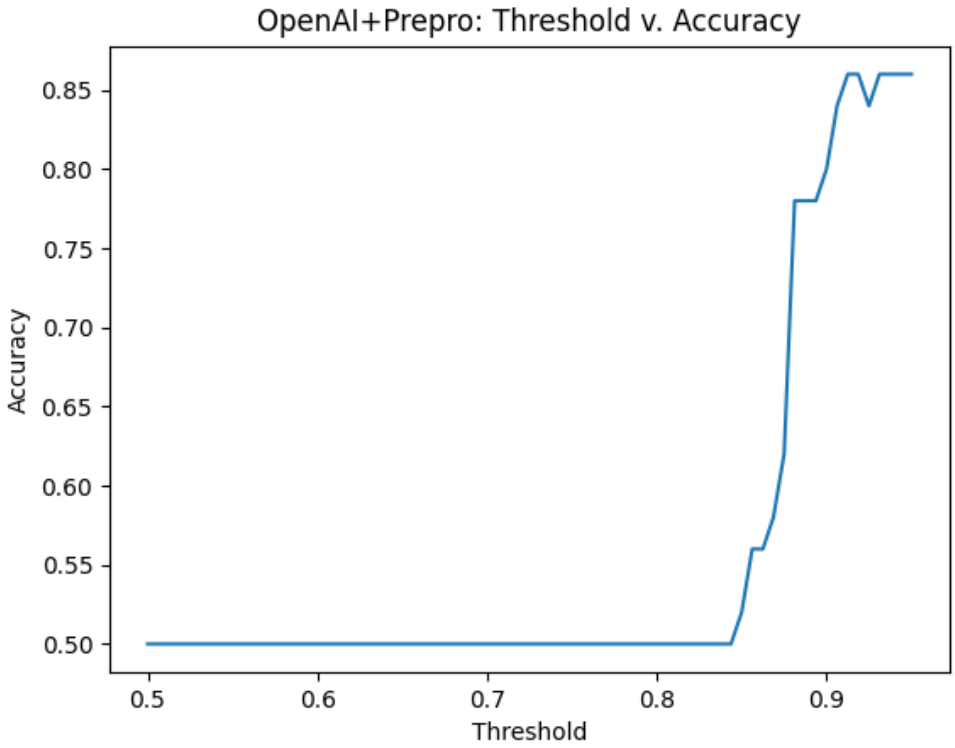

Figure 5. OpenAI+preprocessed: Threshold-Accuracy Plot

There is an interesting behavior in OpenAI's embedding model text-embedding-ada-002 which only performs well in a small range of threshold values. The OpenAI embedding model was trained on a very large data set which probably includes, for example, the entire Wikipedia corpus, and other open-source large dataset in the world wide web. Therefore, OpenAI tries to fit this paper's datasets in its already large vector space containing the OpenAI's large language model dataset (e.g., Wikipedia). Since the 4,400 case decisions are essentially related to each other with similar words and writing style with respect to the possibly millions of documents that OpenAI already knows, the tendency is to produce 4,400 document vectors are very close to each other resulting in smaller distances and higher similarity scores.

In contrast to other embedding models such as Doc2Vec and TF-IDF, these embedding models are only aware of the legal training dataset (4,400 case decisions). Thus, there are higher distances between document vectors. Table 1 below summarizes the accuracy, recall, and specificity using the most optimal similarity threshold identified from Figures 2-5. Generally speaking, all embedding models performed with at least "moderate accuracy" (0.70-0.90). It is worth noting that Doc2Vec achieved the highest accuracy (94%) considered a "high accuracy" (0.90-1.00). However, the accuracy scores for Doc2Vec and the other models are still somehow close to each other, with the largest gap being only 10% between Doc2Vec (94%) and OpenAI+raw (84%). The moderate to high accuracy of the embedding models further reaffirms its ability to capture semantically similar documents, or in this study, judicial cases.

The moderate to high accuracy scores can be attributed to the textual nature of the legal dataset (i.e., Supreme Court case decisions) for having a relatively large word count (approximately 7,900 words on average) which creates identity and character to the legal text, and the consistency of legal keywords (e.g., "murder" as the only legal nomenclature for the crime it describes) which eventually helped in delineating which cases can be grouped together as similar in the vector space during the embedding process.

Table 1. Performance of Embedding Models in Evaluation Measure 1: Similarity Classification

Embedding Model	Threshold (t)	Accuracy	Recall	Specificity
TF-IDF	0.5000	0.84	**0.96**	0.72
Doc2Vec	0.5687	**0.94**	0.88	**1.00**
OpenAI+raw	0.8937	0.88	0.92	0.84
OpenAI+prepro	0.9125	0.86	**0.96**	0.76

4.3 Evaluation Metric 2: Similarity Comparison

Three related studies [21, 2, 26] explored correlation statistics, namely the calculation of Pearson and Spearman correlation coefficients, to assess the strength of the association between cosine similarity scores and evaluations by legal experts in order to evaluate the validity of the embedding model. However, assigning numerical scores by legal domain experts can be a time-consuming, challenging, and potentially confusing task, often rooted in

heavy subjectivity. This subjectivity is especially evident when experts are not provided with a standardized rubric for rating the similarity relatedness of case decision pairs on a scale from 1 (indicating no similarity) to 10 (indicating the highest degree of similarity). Moreover, it is worth noting that two different legal experts or professionals will most likely assign different similarity ratings to the same pair of case decisions. As an alternative, this paper proposes a novel evaluation measure that still captures the fundamental concept of the "similarity ordering" relationship among case decisions. It is more likely that two or more legal experts or professionals could unanimously agree whichever case (*A* or *C*) is more similar with respect to *B*.

For the second evaluation metric "similarity comparison" this study prepared forty-eight (48) "similar triples" that are randomly selected from the legal dataset as shown in Table 2. A "similar triple" is defined as a set of three case decisions that are calculated as semantically similar and therefore, satisfying this condition using the cosine similarity score and a threshold: $(sim(A_i, B_i) > t) \wedge (sim(B_i, C_i) > t) \wedge (sim(A_i, C_i) > t)$ where i is the index of a "similar triple" in the test dataset. Furthermore, the second tests cases are also balanced wherein a set of twelve similar triples are collected from each of the four document embedding models. Then for each similar triples, the legal expert manually determined which is the more similar judicial case A_i or C_i, relative to the judicial case B_i. As mentioned earlier in Section 3.2, the legal expert explained that the similarity of the "legal doctrine/s" is the top priority in determining similarity between judicial cases, followed by similarity in the legal issue/s, and similarity of the fact/s of the case.

The challenge for the embedding is to correctly identify as much as possible which pair (A_i, B_i) or (B_i, C_i) in order to capture the essence of semantic ordering such that there exists a more similar case compared to a similar case. It is one thing to determine if a pair of documents is similar or not, but it is an entirely different task to determine correctly which is more similar among two similar documents which this evaluation metric seeks to measure. The labels of the legal expert is shown completely in Appendix 2.

On the other hand, the predicted labels obtained using the four embedding models and cosine similarity (with respect to the most optimal threshold from Section 3.2 are presented in Table 2. The results indicate that only Doc2Vec has "moderate accuracy" (0.70-0.90) and the three remaining models produced "low accuracy" (0.50-0.70). Doc2Vec is expected to outperform TF-IDF because of unique ability to consider "semantic relatedness." However, while OpenAI company promised that its embedding model has the ability to capture semantic relatedness [28] similar to Doc2Vec, this paper can only hypothesize (due to the confidentiality of OpenAI's embedding implementation) that the issue, again, may be attributed to the compressed document vectors of the legal dataset.

Table 2. Performance of Embedding Models in Evaluation Measure 2: Similarity Comparison

Embedding Model	**True Predicted Labels**	**Accuracy**
TF-IDF	32/48	0.6667
Doc2Vec	**35/48**	**0.7292**
OpenAI+raw	26/48	0.5417
OpenAI+prepro	30/48	0.6250

This is an example of how the embedding models performed in one of the similar triples of the second set of test cases (12th "similar triple" entry). The three cases are the following:

A. Expedition Construction Corp. vs. Africa, G.R. No. 228671 (Dec. 14, 2017)
B. Felicilda vs. Uy, G.R. No. 221241 (Sept. 14, 2016)
C. Vicmar Development Corp. vs. Elarcosa, G.R. No. 202215 (Dec. 9, 2015)

The three cases are described as similar for being labor cases involving a complaint of illegal dismissal. All four embedding models unanimously classified the similar triple as similar to each other and passes the "similarity classification" metric.

A careful reading of these cases reveal nuances in its content. According to the legal expert, the first (A_{12}) and second (B_{12}) cases are "more similar" because the two cases discussed about the requisites to establish an "employer-employee relationship," while the third case (C_{12}) discussed the "regular employment" status of a working group. In this test case entry, Doc2Vec (best performing model) and OpenAI+prepro (third-best performing model) were able to determine that there is higher similarity between *A* and *B.* The two remaining models, TF-IDF and OpenAI+raw, failed to determine this as the "more similar" pair but it is interesting to note the absolute differences between the two similarity scores are small compared to Doc2Vec (see Table 3).

Table 3. Evaluating "Similarity Comparison" in Test Case 12.

	$\boldsymbol{X}$: $\boldsymbol{sim(A_{12}, B_{12})}$	$\boldsymbol{Y}$: $\boldsymbol{sim(B_{12}, C_{12})}$	$\boldsymbol{\lvert X - Y \rvert}$	**Predicted label**
Doc2Vec	**0.7526**	**0.6922**	**0.0604**	$\boldsymbol{sim(A, B)}$
OpenAI+prepro	**0.9414**	**0.9196**	**0.0218**	$\boldsymbol{sim(A, B)}$
OpenAI+raw	0.8954	0.9196	0.0210	$sim(B, C)$
TF-IDF	0.9277	0.9381	0.0103	$sim(B, C)$

The results from this particular sample case is compelling and exhibits how document embedding has potential in accelerating legal research where there is a need to research not only thematically similar cases, but similar cases with strong similarity in terms of doctrine/issue, which is very crucial due to the common-law nature of the Philippine legal system.

V. CONCLUSION AND RECOMMENDATION

In conclusion, this research explored the potential of document embedding techniques in NLP, including Doc2Vec, TF-IDF, and OpenAI (text-embedding-ada-002), in determining similarity between judicial cases that can be used as a document retrieval feature to advance legal research in the Philippines. Additionally, this study also proposed two evaluation metrics in order to analyze and evaluate embedding models, namely "similarity classification" and "similarity comparison." The "similarity classification" metric measures the ability of the embedding model to classify "similar" and "not similar" pairs of cases. On the other hand, the "similarity comparison" metric measures the ability of the embedding model to identify which of the two similar cases is even "more similar" with respect to a judicial case of interest. The four embedding models were tested against the true labels of the legal expert. This research created four different embedding models: (1) TF-IDF, (2) Doc2Vec, (3) OpenAI with raw dataset (i.e., OpenAI+raw), and (4) OpenAI with preprocessed dataset (i.e., OpenAI+prepro), and were evaluated using the proposed evaluation metrics.

The general results indicate that capability of embedding models to expedite legal research and identify related case decisions. In "similarity classification" metric, the four embedding models performed with "moderate accuracy" (0.70-0.90) to "high accuracy" (0.90-1.00) in this task. Doc2Vec achieved the highest accuracy at 94%, followed by the two OpenAI's embedding models at 88% (OpenAI+raw) and 86% (OpenAI+prepro), and lastly, TF-IDF at 84% accuracy. However, in "similarity comparison" metric, all embedding models performed with "low accuracy" (0.50-0.70) with the exception of Doc2Vec who performed with "moderate accuracy" (0.70-0.90) at 72.92% correctly identifying 35 out of 48 similar pairs. The significance of the second evaluation metric becomes more important because it exposes the limitation of embedding model in identifying which of the two cases is even "more similar."

It is interesting to note how Doc2Vec outperformed the state-of-the-art technology such as OpenAI's text-embedding-ada-002 which overwhelmingly has a larger vector size (1,536 dimensions). The OpenAI embedding model liberates the developer from the laborious task of preprocessing and training which demands a certain high standard of hardware specifications. The OpenAI embedding model excelled in the "similarity classification" task nonetheless, but the problem probably lies in its extremely large training data from around the world wide web which tends to fit and squeeze the mere set of this study's legal dataset (4,400 case decisions) clustered closely together and consequently producing closer document vector and similarity scores. This paper hypothesizes that an additional preprocessing step may be necessary for OpenAI embedding such as summarizing all case decisions (case digests) in order to lessen the overall resemblance of the corpus which may increase the variance of their document vectors. On the other hand, this paper suggests that legal tech industries and researchers should move on from using TF-IDF as document vector representation of case documents having the least accuracy score in "similarity classification" test and low accuracy in "similarity comparison." test.

This study highlights the potential of document embedding to improve legal research in the Philippines. Given the low to moderate performance of embedding models in the second evaluation metric ("similarity comparison"), the research and design efforts should be geared

towards improving "similarity comparison" or the ability of the embedding model to identify which is "more similar" than another "similar" than to simply know which is "similar" or "not similar." Future work may involve model fine-tuning techniques and evaluating the model in the context of a multiclass classification problem (e.g., very similar, similar, dissimilar). Due to the popularity of Generative AI, it is also interesting to consider text reorganization (e.g., summaries, sections, etc.) as preprocessing technique through the aid of Large Language Models (LLMs) that provide generate a structured form in unstructured texts such as Philippine Supreme court case decisions. The extension of the training dataset covering the entire case decisions from 1901 until the present time can also be considered for a more accurate classification and comparison task. Furthermore, the study also suggests conducting cross-labelling from multiple legal experts in order to mitigate bias in the true labels. There should be constant collaboration between computer scientists and legal professionals (as stakeholders) in improving legal research through artificial intelligence.

VI. ACKNOWLEDGEMENTS

The author would like to thank Batas.org for their free application released some time back in 2021 during the COVID-19 pandemic which contains the HTML web files of Philippine jurisprudence used as training data set for this study. Furthermore, the author would like to thank Atty. Siegfred G. Perez, a member of the Gerodias Suchiangco Estrella law firm in the Philippines, serving as the legal expert of this study for providing the "true labels" in both test cases, and in enlightening the author more about the practice of law.

References:

[1] Razon BJU, Solano GA, Ran era LTB. 2022. Topic modelling Supreme Court case decisions using latent dirichlet allocation. 13th International Conference on Information and Communication Technology Convergence (ICTC); Jeju Island, Republic of Korea. p. 284-289. doi: 10.1109/ICTC55196.2022.9952945

[2] Ranera LTB, Solano GA, Oco N. 2019. Retrieval of semantically similar Philippine Supreme Court case decisions using Doc2Vec. 2019 International Symposium on Multimedia and Communication Technology (ISMAC); Quezon City, Philippines. p. 1-6. doi:10.1109/ISMAC.2019.8836165

[3] Peramo E, Cheng C, Cordel II M. 2021. Juris2vec: Building word embeddings from Philippine Jurisprudence. 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC); Jeju Island, Republic of Korea. p. 121-125. doi: 10.1109/ICAIIC51459.2021.9415251

[4] Virtucio MBL, Aborot JA, Abonita, JKC, Avinante RS, Copino RJB, Neverida MP, Osiana VO, Peramo EC, Syjuco JG, Tan GBA. 2018. Predicting decisions of the Philippine Supreme Court using natural language processing and machine learning. IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC); Tokyo, Japan. p. 130-135. doi: 10.1109/COMPSAC.2018.10348

[5] Martija MA, Domoguen J, Naval P. 2019. How deep is your law? Predicting associations between cases in Philippine Jurisprudence. TENCON 2019-2019 IEEE Region 10 Conference (TENCON); Kochi, India. p. 886-891. doi: 10.1109/TENCON.2019.8929425

[6] Rosales MA, Magsumbol JV, Palconit MGB, Culaba AB, Dadios EP. 2020. Artificial intelligence: The technology adoption and impact in the Philippines. 2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and

Management (HNICEM); Manila, Philippines. p. 1-6. doi: 10.1109/HNICEM51456.2020.9400025
[7] Supreme Court Public Information Office. 2022. Retrieved from https://sc.judiciary.gov.ph/sc-to-use-artificial-intelligence-to-improve-court-operations/ on 10 Sept 2023.
[8] Supreme Court Public Information Office. 2023. Retrieved from https://sc.judiciary.gov.ph/chief-justice-gesmundo-sc-to-use-ai-powered-tools-to-improve-court-legal-research/ on 10 Sept 2023.
[9] Supreme Court Public Information Office. 2023. Retrieved from https://sc.judiciary.gov.ph/chief-justice-gesmundo-judiciary-e-library-to-use-ai-technology-to-improve-legal-research/ on 10 Sept 2023.
[10] Liddy ED. 2001. Encyclopedia of Library and Information Science. 2nd ed. New York: Marcel Decker, Inc.
[11] Keet S, Ayesha B, de Silva N, Perera AS, Jayawardana V, Lakmal D, Perera M. 2019. Legal document retrieval using document vector embeddings and deep learning. Intelligent Computing: Proceedings of the 2018 Computing Conference. 2:160-175.
[12] Le Q, Mikolov T. 2014. Distributed representations of sentences and documents. International Conference on Machine Learning. p. 1188-1196.
[13] Dai AM, Olah C, Le QV. 2014. Document embedding with paragraph vectors. NIPS Deep Learning Workshop.
[14] Chib V, Jafri A. 2019. An Effective approach of extracting local documents from the distributed representation of text using document embedding and latent semantic analysis. 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT); Tirunelveli, India, 2019. p. 152-156. doi:10.1109/ICSSIT46314.2019.8987859
[15] Philippine Statistics Authority. 2021. Retrieved from https://psa.gov.ph/content/2020-census-population-and-housing-2020-cph-population-counts-declared-official-president.
[16] Asean Today. 2017. Retrieved from https://www.aseantoday.com/2017/02/philippine-judiciary-and-criminal-justice-system-under-pressure-an-inside-look on 15 Sept 2023.
[17] Philippine Daily Inquirer. 2023. Retrieved from https://newsinfo.inquirer.net/1732856/fwd-tulfo-stresses-need-for-more-lawyers-questions-ched-for-insufficient-budget-of-leb on 15 Sept 2023.
[18] Thongtan T, Tanasanee Phienthrakul T. 2019. Sentiment classification using document embeddings trained with cosine similarity. 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. p. 407-414.
[19] Zhang Y, Chen Q, Yang Z, Lin H, Lu Z. 2019. BioWordVec, improving biomedical word embeddings with subword information and MeSH. Scientific Data. 6 (52): 1-9. https://doi.org/10.1038/s41597-019-0055-0
[20] Chalkidis I, Kampas D. 2019. Deep learning in law: early adaptation and legal word embeddings trained on large corpora. Artificial Intelligence and Law. 27: 171–198. https://doi.org/10.1007/s10506-018-9238-9
[21] Mandal A, Chaki R, Saha S, Ghosh K, Pal A, Ghosh S. 2017. Measuring similarity among legal court case documents. 10th Annual ACM COMPUTE Conference; India, 2017. p. 1-9.
[22] Almuslim I, Inkpen D. 2020. Document level embeddings for identifying similar legal cases and laws (AILA 2020 shared Task). Forum for Information Retrieval Evaluation; Hyderabad, India. p. 42-48.
[23] Bhattacharya P, Ghosh K, Pal A, Ghosh S. 2022. Legal case document similarity: You need both network and text. Information Processing & Management. 59(6):103069.
[24] Kumar S, Reddy PK, Reddy VB, Singh A. 2011. Similarity analysis of legal judgments. ACM Compute Conference. 17:1-4.
[25] Kumar S, Reddy PK, Reddy VB, Suri M. 2013. Finding similar legal judgements under common law system. 8th International Workshop Databases in Networked Information Systems 2013.
[26] Mandal A, Ghosh K, Ghosh S, Mandal S. 2021. Unsupervised approaches for measuring textual similarity between legal court case reports. Artificial Intelligence and Law. p. 1-35.
[27] Novotna T. 2020. Document similarity of Czech Supreme Court decisions. Masaryk University Journal of Law and Technology. 14(1):105-122.
[28] OpenAI. 2022. Retrieved from https://openai.com/blog/new-and-improved-embedding-model on 19 Apr 2024.
[29] Aizawa A. 2003. An information-theoretic perspective of tf–idf measures. Information Processing and Management. 39:45–65.
[30] Ramos J. 2003. Using TF-IDF to determine word relevance in document queries. First Instructional Conference on Machine Learning. 242(1): 29-48.
[31] Kumar S, Musharaf D, Sagar AK. 2023. A comprehensive review of the latest advancements in large

generative AI models. Communications in Computer and Information Science. Cham: Springer Nature Switzerland. p. 90-103.

[32] Li X, Henriksson A, Duneld M, Nouri J, Wu Y. 2023. Evaluating embeddings from pre-trained language models and knowledge graphs for educational content recommendation. Future Internet. 16(1):12.

[33] OpenAI. Retrieved from https://openai.com/blog/introducing-text-and-code-embeddings on 19 Apr 2024.

[34] Arvix. 2022. Retrieved from https://arxiv.org/pdf/2201.10005.pdf on 19 April 2024.

[35] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. 2017. Attention is all you need. 31st Conference on Neural Information Processing Systems (NIPS 2017).

[36] Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J. 2020. Transformers: State-of-the-art natural language processing. 2020 conference on Empirical Methods in Natural Language Processing: System Demonstrations. p. 38-45.