# Solving Bar Exam Questions with Deep Neural Networks

Adebayo Kolawole John
Department of Computer Science, University of Torino
Corso Svizzera 185
Torino, 10149, Italy
collawolley3@yahoo.com

Luigi Di Caro
Department of Computer Science, University of Torino
Corso Svizzera 185
Torino, 10149, Italy
dicaro@di.unito.it

Guido Boella
Department of Computer Science, University of Torino
Corso Svizzera 185
Torino, 10149, Italy
guido@di.unito.it

## ABSTRACT

In this paper, we present a system which solves a Bar Examination written in Natural Language. The proposed system exploits the recent techniques in Deep Neural Networks which have shown promise in many Natural Language Processing (NLP) applications. We evaluate our system on a real Legal Bar Examination, the United States Multi-State Bar Examination (MBE), which is a multi-choice 200-questions exam for aspiring lawyers. We show that our system achieves good performance without relying on any external knowledge. Our work comes with an added effort of curating a small corpus, following similar question answering datasets from the well-known MBE examination. The proposed system beats a TFIDF-based baseline, while showing a strong performance when modified for a legal Textual Entailment evaluation.

## 1. INTRODUCTION

Many tasks in Natural Language Processing (NLP) involve generation of semantic representation for proper text understanding. For example, tasks like Textual Entailment [5] and Question Answering [11, 31] involve deep semantic understanding of the text since a popular approach like the Bag of Words (BOW) has limitations due to natural language ambiguity.

Question Answering (QA) tasks follow the Human learning and testing process. For instance, a student reads a course note in order to obtain some facts and background knowledge. The student then answers any question based on the facts available to him. This is the main essence of learning, which is about 'committing to memory' and 'generalizing' to new events. Even though learning seems to be a natural phenomenon to humans, it is nevertheless still a challenging goal for computers to replicate. Researchers working in the Computer Science field of Machine Learning (ML) often employ methods to analyze existing data in order to predict the likelihood of uncertain outcomes. These methods usually produce results that approximate human capabilities [19].

The term *ML* is actually a broad term used to describe supervised or unsupervised approaches for making the computer identify patterns in our data. Usually, a human handcrafts some features from the data, and the extracted features are then shown to the algorithm for it to learn the latent discriminating features. Finally, the algorithm learns to predict the outcome of an unseen event. Neural Networks (NN) [8] are now extensively used by researchers because they offer a higher representational power. NN try to mimic the cognitive system of the human. They have a lot of interconnected nodes. Each node receives some inputs from the lower layer nodes, performs a computation on the input by using some non-linear functions, and lastly, the node transmits its output to the nodes in the layer above it. Such a network with many interconnecting layers stacked is called a *Deep Neural Network* (DNN) [24].

When performed by a human, QA requires some form of cognitive abilities such as reasoning, meta-cognition, the contextual perception of abstract concepts, intelligence, and language comprehension. Although machines are yet to replicate a strong cognitive ability like a human, nevertheless, the non-cognitive computational techniques that employ heuristics and statistical approximation can rightly model most problems while giving an 'intelligent' result which is close to that from a human [27]. We leverage this assumption by taking for granted the cognitive capability comparison to our system. Instead, the goal is to achieve a result that is presumed acceptable by a human examiner.

In the QA task, systems are provided with a text passage containing some facts or background knowledge, and a question which is related to that text passage. Furthermore, an answer to the question is provided. The system is then given a similar but slightly different question and is expected to answer it from the same background knowledge.

The remaining part of the paper is organized as follows. In the next section, we review the related work. This is followed by a description of the MBE Exam and the corpus used for the experiment. Next, we describe our approach. Finally, we describe the experiment and evaluation.

## 2. RELATED WORK

NNs have shown good performance in many NLP tasks including QA. The authors in [31, 12] achieved an excellent result with DNN for QA. In particular, [31] achieved 100% accuracy on some tasks.[1] Similarly, the work of [26] and the *Answer-Sentence Selection* proposed by Feng [10] are also based on NN. A considerable portion of the QA systems use

In: Proceedings of the Second Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2017), June 16, 2017, London, UK.

Published at http://ceur-ws.org

[1] e.g. the single supporting facts and two supporting facts on BaBi dataset. A similar result was reported for CBT and Simple Question datasets. The datasets are accessible at https://research.facebook.com/research/babi/

a synthetic dataset. For example, the dataset in [31] was generated by simulating time-stepped facts using entity, location and temporal information, e.g.,

**Ex** 1:
1. James is watching TV in his bedroom
2. James is Sleeping
3. Where is James? -bedroom

The models in [31, 12, 26] were trained to memorize factual information about the entities in a given story, e.g., keeping track of the *where*, *when*, and *who* information regarding an entity. Furthermore, the questions are quite simple. Each question requires only a factoid answer. According to the authors, it is expected that a question should be unambiguous [31, 13]. Bordes et. al., [3] utilized a more challenging dataset. Nevertheless, the questions still require factoid answers. In particular, the dataset contains *list* questions, i.e., a question with multi-choice answers. The work in [12, 13, 31] showcases an array of experiments which is aimed at examining and estimating the text comprehension capability of a QA system.

Some QA systems exploit external information, i.e., those available in a knowledge base, a Semantic Net, or the Internet, for generating a plausible answer to a question. For instance, some researchers utilized a collection of facts which have been extracted from a large text collection in form of *Subject-Relation-Object* (SVO) triples. The triples are then stored in a knowledge base [7, 6]. The QA system is therefore trained to map a question to the relevant fact in the knowledge base. This often requires transcribing a question into a format that can easily be matched to the fact in the knowledge base. The problem with this approach is the over-reliance on a structured set of facts, e.g., (Donald Trump, is-president-of, United States). Moreover, the SVO triples may be difficult to curate, the triple extraction algorithms may overgenerate, and the accuracy for SVO extraction may not be optimal. Also, there is presently no domain-specific collection of SVO fact triple for the Legal domain.

A few QA systems address solving a real exam question. The closest to our work in this regard is QANTA [15] which learns word and phrase-level representations with a Recurrent Neural Network (RNN) for identifying an answer that appears as an entity in the paragraph. The authors in [2] presented a system for solving biology questions. Similarly to QANTA, the paragraphs contain a description of a biological process, a short question, and two choice answers out of which only one is the correct answer. Weston et. al., [32, 31] employed a Memory Network for the *BAbI* tasks.[2] The BAbi task includes the single-supporting fact and multiple-supporting facts. However,some of the supporting facts are irrelevant to the answer. Also included are the yes/no questions, and list/set questions. The Memory Network follows the Long Short-Term Memory (LSTM) which is a NN that is capable of retaining information over a longer time-step than a typical RNN. The McTest challenge proposed by Yin et. al., [35] is also very related to our work. The essential differences are the nature of the data used, the long sequences of both paragraphs, question, and answers in our dataset, as well as the format that the MBE exam question takes.

However, there is limited prior work in the legal domain in this respect. Most of the reviewed systems require a factoid answer. Furthermore, the datasets are mostly synthetic datasets, i.e., not a real examination question and answer. It is a popular saying that the '*Language of Law*' does not follow the '*Law of Language*'. This is because being domain specific, legal texts employ legislative terms. For instance, a sentence may reference another sentence (e.g., an article) without any explicit link. Also, sentences are generally long and often come with several clausal dependencies. Moreover, there is usually a couple of inter- and intra-sentential anaphora resolution that must be resolved. Wyner [33] lists several NLP issues regarding the legal domain.

The authors in [18, 17] employed a collection of legal text. The dataset[3] was indeed prepared from the Japanese Bar Examination. The task was proposed as a Textual Entailment (TE) task. The dataset consists of Japanese Civil Code articles, some of which were used as the *premise t*, and others the *hypothesis h*. The authors utilized a couple of handcrafted features which are similar to the BOW features usually employed for text similarity and IR. Similar work was done in [29], where the authors mined reference information from a collection of legal text.

The most related work to ours is the work of Biralatei et. al., [9] which makes use of a real legal examination question set. Specifically, the authors make use of the USA Multi-State Bar Examination (MBE). In their experiment, they use 100 real multi-choice answer-question sets. Since each question has *4* available answers out of which only one is correct, they proposed a TE solution. By performing a transformation on a question and each corresponding answer, they obtained 400 $t$ and $h$ pairs, where $t$ is the background knowledge giving as the text passage to a question, and $h$ is a transformed question-answer output. More explicitly, the transformed question-answer output is a combination of a question and a possible answer. Consequently, the authors aimed to see if the transformed text is entailed by a *passage*. Analogous to the work described in [18], the proposed TE system heavily profits from some handcrafted features which typify a similarity between $t$ and $h$.

However, handcrafting a feature is an expensive and time consuming process. It is easy to have noisy features and a series of ablation test is required to identify the best features. Also, their approach relies on word-similarity and synonym substitution using existing knowledge resources like WordNet and VerbOcean. The authors then compute a BOW-based similarity feature between $t$ and $h$. The problem with this approach is that the BOW-based approaches usually suffer from language ambiguity.[4] Furthermore, the approach assumes that a text passage will have a lot of word overlap with the transformed $h$ in case there is an entailment. This assumption is costly and may not hold at all times. Moreover, some questions require extra knowledge apart from what can be explicitly deduced from the given passage.
The following example expatiates this point,

**Example** 2:
**Passage**: A truck driver from State A and a bus driver from State B were involved in a collision in State B that injured the truck driver. The truck driver filed a federal diversity

[2] BaBi dataset is available at https://research.fb.com/projects/babi/

[3] Released as part of the COLIEE Legal IR challenge. http://webdocs.cs.ualberta.ca/~miyoung2/COLIEE2016/

[4] e.g. synonymy, polysemy etc.

action in State B based on negligence, seeking $100,000 in damages from the bus driver.
**Question**: What law of negligence should the court apply?

- **Answer A (false)**: The court should apply the federal common law of negligence.

- **Answer B (false)**: The court should apply the negligence law of State A, the truck driver's state of citizenship.

- **Answer C (false)**: The court should consider the negligence law of both State A and State B and apply the law that the court believes most appropriately governs negligence in this action.

- **Answer D (true)**: The court should determine which state's negligence law a state court in State B would apply and apply that law in this action.

In example *2* above, the *passage* represents the context or some knowledge needed for answering the question. Given this example, an entailment-based system which focuses on similarity would fail since answering the question requires not just the word overlap but an understanding of the semantics of the underlying texts.

This work seeks to address this issue by proposing a NN Legal Question Answering (LQA) system which employs a LSTM to encode and decode the question-answer pair for a good semantic representation. A LSTM is a type of RNN with slightly more powerful language modeling capacity and it has become one of the most successful methods for end-to-end supervised learning. Furthermore, LSTMs exhibit a memory bank property since they are able to retain information over many time-steps while also overcoming the vanishing gradient problem [14, 32, 3].

Our goal is to evaluate how well the proposed approach can perform on a legal text reasoning task, and if the performance of our model can compete with that of a human. Generally, MBE examinees are required to correctly pass at least 125 out of the 200 standard MBE questions. Although the 125 score benchmark is not absolute, an examinee is also required to get a certain number of points from the essay exam. We assume that our model competes if it obtains a score that is above the MBE nationwide *Mean* score, which is computed based on statistical analysis of past MBE examinations. Table 1 shows the summary statistic of the national performance for the year 2016.[5] The Maximum score obtained is 188/200, which is around 94%. The Minimum is 58/200, which is about 29%, and the Mean score is 143/200, which is approximately 71.5%. We also introduce a new Legal QA corpus, specified in two formats which we describe in the subsequent section, and thereby propose a new form of Legal Question Answering task.

Many people from outside the ML field often regard NNs as black-box whose performance cannot be analyzed. To assuage this sentiment, we benchmark our system against a TFIDF baseline which predicts its outcome based on a TFIDF similarity between the passage, question, and answer in a way similar to the TE setting of [9]. By obtaining a significantly better result than the baseline, we validate the performance of our system.

| | **Feb (2016)** | **July (2016)** | **Total (2016)** |
|---|---|---|---|
| **Min Score** | 72.5 | 58.6 | 58.6 |
| **Max Score** | **188.2** | **187.4** | **188.2** |
| **Mean Score** | 135.0 | 140.3 | 143.5 |
| **Median Score** | 135.2 | 140.8 | 138.6 |
| **Standard Dev** | 15.0 | 16.7 | 16.4 |
| **No of Examinees** | 23,324 | 46,518 | 69,842 |

**Table 1: 2016 MBE National Summary Statistics (Based on scaled scores). Note: The values reflect valid scores available electronically as of 1/18/2017**

# 3. THE LQA CORPUS

For a human to answer a question, he has to have some facts about the question. We can then generally make deductions using the facts as well as some background knowledge in order to provide a plausible answer. The question answering task mimics this simple approach whereby a background knowledge from which to infer facts is provided. A question is then given and an examinee has to make a judgment using these facts. Some questions can be direct, such that the expected answer is straightforward. E.g., someone who has access to a book on current affairs can easily answer a question like '*who is the president of the USA*?' -Donald Trump. However, some questions require more than a set of facts for someone to be able to answer them correctly. This type of question requires logic in order to make a deduction from the available facts. A typical example is the Bar examination.

The MBE is a *six*-hour, *200*-questions multiple-choice examination developed by the National Conference of Bar Examiners (NCBE), and administered by the user jurisdiction as part of the Bar Examination. The goal of the exam is to assess the extent to which an examinee can apply fundamental legal principles and legal reasoning in order to analyze a given fact pattern.[6] The exam is very important for it is one of a number of measures that the NCBE may use in determining an aspiring lawyer's competence to practice.
Each data point in the exam is a tuple, S = (P, Q, $A_1^4$). Here, *P* is the passage or background knowledge, *Q* is the question, and *A* is the answer. Since it is a multi-choice exam, there are four possible options in *A*, out of which only one is correct and must be selected as the answer. The exam covers a wide area of law including Constitutional Law, Contracts Law, Criminal Law, Evidence, Real Property, Torts, and Civil Procedure.

Similar to the approach in [9], for each *A*, we also split *S* such that we have a separate representation for (P,Q,$A_i$). However, since our goal is not a Textual Entailment task, we ignore any transformation on the text to obtain a t-h pair as it is the case in [9]. In our case, each question-answer sample *S* is represented as 4 mini samples, i.e., $s_1$, $s_2$, $s_3$, $s_4$ such that each s is also a 4-tuple (P, Q, $A_i$, F). Where P,Q,A remains the same and F symbolizes a binary flag for identifying whether the answer is correct or not. In other words, the goal is to determine if a specific answer is suitable to a question, given a background knowledge. The task is then formalized as an Answer-Sentence-Selection task.

**Example** 3:
**Passage**: An entrepreneur from state A decided to sell hot sauce to the public, labeling it 'Best Hot Sauce'. A company incorporated in state B and headquartered in state

[5] http://www.ncbex.org/publications/statistics/mbe-statistics/

[6] http://www.ncbex.org/exams/mbe/

C sued the entrepreneur in federal court in state C. The campaign sought $50,000 in damages and alleged that the entrepreneur's use of the name 'Best Hot Sauce' infringed the company's federal trademark. The entrepreneur filed an answer denying the allegations, and the parties began discovery. Six months later, the entrepreneur moved to dismiss for lack of subject-matter jurisdiction.

**Question**:Should the court grant the entrepreneur's motion?

1. **Answer A (True)**: No, because the complaint's claim arises under federal law.
   - **Evidence**: The claim asserts federal trademark infringement, and therefore it arises under federal law. Subject-matter jurisdiction is proper under 28 U.S.C. $1331 as a general federal-question action. That statute requires no minimum amount in controversy, so the amount the company seeks is irrelevant.
   - **Label**: 1
2. **Answer B (False)**: No, because the entrepreneur waived the right to challenge subject-matter jurisdiction by not raising the issue initially by motion or in the answer.
   - **Evidence**: Under Federal Rule 12(h)(3), subject-matter jurisdiction cannot be waived and the court can determine at any time that it lacks subject-matter jurisdiction. Therefore, the fact that the entrepreneur delayed six months before raising the lack of subject-matter jurisdiction is immaterial and the court will not deny his motion on that basis.
   - **Label**: 0
3. **Answer C (False)**: Yes, because although the claim arises under federal law, the amount in controversy is not satisfied.
   - **Evidence**: There is no amount-in-controversy requirement for actions that arise under federal law.
   - **Label**: 0
4. **Answer D (False)**: Yes, because although there is diversity the amount in controversy is not satisfied.
   - **Evidence**: Federal Rule 4(e)(2) governs service on individual defendants and authorizes service on a person of 'suitable age and discretion' only when service is made at the defendant's dwelling or usual place of abode, not at the defendant's workplace.
   - **Label**: 0

Example *2* shows a sample passage and the corresponding question and answers. We can see that the option labeled as 'True' is the only correct answer.

The second format takes a similar style. However, we introduce extra knowledge in the form of an explanation made by an expert to validate why an answer is correct or not. Each sample is thus a 5-tuple (P, Q, A$_i$, E, F). Where P,Q,A,F remains the same and $E$ symbolizes the extra knowledge which justifies $F$. We say that $E$ is the evidence since it justifies or explains why an answer is said to be correct or incorrect. Example *3* shows the passage, question, answer along with the *evidence* which explains why the answer is correct or wrong. The goal is to make the system take advantage of the extra knowledge since many questions cannot be directly inferred from the passage without an extra information. It can be seen in example *3* that there is an absence of clear linguistic overlap between the passage text and the answer text. Also, the passage text contains less or no information required for answering a question. In this scenario, an extra information (evidence) may indeed be helpful for answering the question.

For the purpose of LQA corpus, we use a random sample of 550 out of the 600 available passage-question-answer set from the 1991 MBE-I, 1999-MBE-II, 1998-MBE-III and some exam practice samples obtained from the examiner.[7] We choose to use the exam questions because they are publicly available and have a gold standard answer. We prepared the question set in the (P, Q, A$_i$, F) format explained earlier, yielding 2200 passage-question-answer-flag.[8] For the second format with extra knowledge $E$, we obtained 15 annotated passage-question texts. In total, we obtained a set of 60 question-answer sets in (P, Q, A$_i$, E, F) format. Because the number seems quite small, we are working towards getting annotations for more samples. We rely on the validity/correctness of the gold standard and annotations obtained from our sources.

## 4. NEURAL REASONING OVER LQA

Recently, NN algorithms such as the RNN [20] and LSTM [14] have excelled at language modeling tasks. LSTM, a variant of RNN, is especially powerful since it is robust to the vanishing gradient problem and has a memory that is controlled by the input gate, the forget gate, and the output gate. The LSTM is therefore, able to retain information over several time steps, i.e., a long sequence of words.

LSTMs have been deeply studied [14, 28] and have variants like the Memory Networks [32, 31] which is specifically wired to retain information over longer sequences. A LSTM network learns short and long-range contextual information.

At each time step t, let an LSTM unit be a collection of vectors in $\mathbb{R}^d$ where $d$ is the memory dimension: an *input gate* i$_t$, a *forget state* f$_t$, an *output gate* o$_t$, a *memory cell* c$_t$ and a *hidden state* h$_t$. The u$_t$ is a tanh layer that applies a non-linear function to the received input and creates a vector of new candidate values that could be added to the state. The state of any gate can either be open or closed, represented as [0,1]. The LSTM transition is represented by the following equations. (x$_t$ is the input vector at time step $t$, $\sigma$ represents the sigmoid activation function, and $\odot$ is the element-wise multiplication) :

$$i_t = \sigma\Big(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}\Big),$$

$$f_t = \sigma\Big(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}\Big),$$

$$o_t = \sigma\Big(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\Big),$$

[7] http://www.ncbex.org/exams/mbe/

[8] Our corpus is available on request

$$u_t = \tanh\left(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}\right),$$

$$c_t = i_t \odot u_t + f_t \odot c_{t-1},$$

$$h_t = o_t \odot \tanh c_t \tag{1}$$

# 5. METHODS

We describe the general framework of our model in this section. Given a set of inputs, the goal is to find an input representation that encodes both the passage $P$, the question $Q$, and the answer $A$. Our model is essentially a distributional sentence model which is able to comprehend the semantics of the input texts. Our model has three key components, i.e., the encoder module, the interaction module, and the output module.

## 5.1 Input Encoder

At the input layer, we introduce three bi-directional LSTM (BiLSTM) encoders that read the sequences of $P$, $Q$, and $A$ separately. A BiLSTM is essentially composed of two LSTMs. One capturing information in one direction from the first time step to the last time-step while the other captures information from the last time-step to the first. The outputs of the two LSTMs are then combined to obtain a final representation. Here, we represent each word in the sentences P, Q and A with a *d*-dimensional vector, where the vectors are obtained from a word embedding matrix. Generally, we use the *Glove* 300-dimensional vectors obtained after training the Glove algorithm on 840 billion words [23]. In practice, a domain-specific embedding can be learned from a collection of legal texts by using an algorithm like Word2Vec [21]. However, our dataset is quite small for any useful embeddings to be generated with the Word2Vec algorithm. While building the vocabulary, any citation of a Law article, e.g, *2.8 U.S.C .& 1331*, date or money, e.g., *$50,000* in a text is represented by a special symbol. Also, entities such as *State A*, *State B* or *State C* are automatically identified and given a special symbol. Each special symbol in the vocabulary is associated with a randomly initialized vector in the embedding matrix. We encode and obtain the sentence representation of each input text using equation 2 such that a vector representation that captures the meaning of each text is learned:

$$\overrightarrow{h_i} = LSTM(\overrightarrow{h_{i-1}}, P_i), \quad i \in [1, ..., M]$$

$$\overleftarrow{h_i} = LSTM(\overleftarrow{h_{i-1}}, P_i), \quad i \in [M, ..., 1]$$

$$BiLSTM(P) = [\overrightarrow{h_i}; \overleftarrow{h_i}]$$

$$h_p = BiLSTM(P) \tag{2}$$

## 5.2 Interaction Layer

The interaction layer is formalized as a hierarchical attention layer for reducing the input space from three to two. Attention is a way of focusing on some important parts of an input, and has been used extensively in some language modeling tasks such as machine translation, natural language inference and document classification [1, 22, 34]. Essentially, it is able to identify the parts of a text that are most important to the overall meaning. We use two forms of attention, namely *inter* and *intra* attention. The *intra* attention focuses on the important words within the same text. Specifically, such important words can now be aggregated to compose the meaning of the text. The implication of this is that we can use the intra attention to focus on important words independently for each $P$, $Q$, and $A$ text. On the other hand, the *inter* attention tries to attend to the important words in one text conditioned on the intra-attention weighted representation of the second text. Analogously, the inter attention allows for an interaction between two texts and ensures that we focus on words that are most important for representing the meaning of one text, in the context of the other text.

Following [1], we use intra-attention to obtain the sentence representation as shown in equation (3). Initially, the encoded sentence (see equation (2)) is first passed through a Multi-Layer Perceptron (MLP) Neural Network to get a hidden representation $\mathrm{u}_i$ which is then weighted with the attention vector $\alpha_i$ across the time steps. The attention vector $\alpha_i$ is implemented as a Softmax whose weights sum up to 1, and are used to compute a weighted-average of the last hidden layers generated after processing each of the input words.

$$u_i = \tanh(W_p h_p + b_p)$$

$$\alpha_i = \frac{\exp(u_i^M u_p)}{\sum_p \exp(u_i^M u_p)}$$

$$h_s = \sum_p \alpha_i h_i \tag{3}$$

Here, $i$ signifies each time step in $h_p$. $h_p$ is the encoded text, and M is the number of time-steps in $\mathrm{h}_p$. The vector $\mathrm{u}_p$ is a context vector which may be randomly initialized.

The inter attention follows a similar approach. In particular, we use it to capture the interaction between the sentences using equation (4). Specifically, what this means is that we can use one inter attention layer to obtain the interaction between the intra-attention hidden states of the encoded passage text and that of the encoded question text (P $\rightarrow$ Q). Also, the same attention layer is employed to capture the interaction between the encoded question text and encoded answer text (Q $\rightarrow$ A). Each of the interactions generated with the inter-attention produces a high-level representation of these texts which can now be used for classification. Put in another way, we obtain two vectors which summarize the interaction between the input sentences.

$$u_s = \tanh(W_s h_s + b_s)$$

$$\alpha_s = \frac{\exp(u_s^N u_q)}{\sum_q \exp(u_s^N u_q)}$$

$$s = \sum_q \alpha_s h_s \tag{4}$$

## 5.3 Output layer

The task can be simplified as a binary classification task since an answer either has a label 0 or 1. Because the two vectors $\mathrm{s}_p$ and $\mathrm{s}_q$ are the ensuing representations which can be regarded as the high-level representation of the interaction between texts P, Q and A. In supervised learning, when there is a sufficient number of positive and negative samples for a category of example, we can formalize the task as a ranking task, trying to create a margin between the positive and negative examples, and ranking based on the margin. There are different approaches to the *Learning to Rank* task, e.g., *Pointwise*, *Pairwise*, and *Listwise* [4]. A Pointwise ranking is straighforward and involves training a binary classifier, i.e., given a triple of question $q$, answer $a$, and a label $y$, as (q$_i$, a$_{ij}$, y$_{ij}$), the ranking function is given as h(**w**, $\psi$ (q$_i$, a$_{ij}$)) $\Rightarrow$ y$_{ij}$. Here, the $\psi$ function creates a feature vector from the question and answer sample. Also, **w** is a vector of model weights.

In order to implement our binary classifier, we concatenate the vectors s$_p$ and s$_q$ (see (5)) and then propagate the output of the concatenation to a MLP where the interaction is fully modeled. Finally, a Softmax layer is used to distribute the probability over the labels.

$$s_{concat} = [s_p; s_q] \tag{5}$$

Formally, we denote l$_i$, i = 1, 2, 3, ..., N-1 as the intermediate hidden layers, W$_i$ as the i-*th* weight matrix, and b$_i$ as the i-*th* bias term. The hidden layer computation of the MLP can be represented as follows:

$$l_1 = W_1 s_{concat}$$

$$l_i = f(W_i l_{i-1} + b_i), \quad i = 2, 3, ...., N-1$$

$$y_o = f(W_N l_{N-1} + b_N) \tag{6}$$

where $y_o$ is the output vector of the last layer, $f$ is a non-linear function which, in this work, is the hyperbolic tangent (tanh) activation function, and N represents the number of layers in our neural network. The predicted class is obtained by passing the output vector $y_o$ through a softmax layer as shown in equation (7).

$$\hat{y} = Softmax(W_c y_o + b_c) \tag{7}$$

where $y_o$ is the output vector from the outermost tanh layer, W$_c$ and b$_c$ are the weight matrix and bias vector which are the parameters to be learned by the network, and Softmax is a non-linear activation function that distributes the class probabilities as shown in equation 8. $\hat{y}$ is the predicted class.

$$Pr(\hat{y} = c|y) = \frac{e^{y\theta_c}}{\sum_{k=1}^{K} e^{y\theta_k}} \tag{8}$$

where $\theta_k$ is the weight vector of the k-*th* class.

# 6. SYSTEM EVALUATION

We now describe the experiment and the result obtained. Recall that the goal of our model is to identify whether an answer is correct given a question and a corresponding passage. This is different from the TE task which seeks to establish whether the hypothesis can be inferred from the premise.

## 6.1 Training Parameter

We implemented our model inspired by the work in [26, 32]. As we have mentioned earlier, instead of encoding both the passage, question and answer text sequences as one-hot encoding representations of the token sequences, we used the pretrained 300-dimensional GloVe vectors [23]. We keep the embedding weights fixed throughout the training. The embedding vectors are obtained from an algorithm which is based on the distributional hypothesis [30]. The algorithm operates such that, when given the contexts of a word, it is able to predict words that may appear close to that word. It turns out that it captures many semantics characteristics of a text, such as similarity and relatedness. It has been widely applied in numerous NLP tasks. We use the Keras[9] Deep Learning library to prototype our model. The training data is usually split into 80:10:10 for training, evaluation, and test respectively. We uniformly use a dropout of 0.20, a batch size of 8, ADAM optimizer and a learning rate of 0.01. The model was trained for 20 epochs. Even though we already apply dropout [25] throughout the model, we also use early stopping to avoid over-fitting, usually stopping the training after 4 consecutive epochs without any drop in the validation loss. The model used for testing is the best obtained with the validation set. We found out that our best model is achieved by epoch 10 after which, if we continue to train, we keep getting very high accuracy on the training data which does not generalize to the validation and test set.

| System Description | (Accuracy %) |
|---|---|
| TFIDF-based | 44.80 |
| Our Model | **71.90** |

**Table 2: Standard Evaluation on LQA dataset**

| Human Performance | (Accuracy %) |
|---|---|
| Minimum | 0.29 |
| Mean | 71.50 |
| Maximum | **94.00** |
| **This paper** | 71.90 |

**Table 3: Comparison with human performance (2016 NBEX national statistics)**

## 6.2 Experiment

We compare the results of our model against a TFIDF baseline. The TFIDF baseline is based on a simple assumption that, if we consider the TFIDF scores of the passage text (i.e., $P$) on one hand, and the question and answer texts (i.e., $Q$ + $A$) on the other hand, a high similarity between the TFIDF scores indicate relevance of the answer to the question. The TFIDF feature of ($Q$ + $A$) is subtracted from the TFIDF feature of ($P$) and the resulting vector is passed through a MLP along with the label. This is a simple MLP classification approach. This is a naive assumption, however, we consider it an adequate baseline. Specifically, we would like to know whether our model is capturing only word overlap features or actual logic in form of the semantic of a text. Intuitively, we expect the TFIDF-based model to capture overlap features. However, a good system must demonstrate that it captures not just the word overlap features but also, the semantics and other legal nuances in a text. Table 2 shows the result obtained from the experiment. The table shows a comparison of the performance of our model to a TFIDF-based predictor. Table 3 shows a comparison of our model to the student performance in the MBE exam in the year 2016.

In order to allow for comparison with a few legal TE systems, we modify our model such that the input space is

[9] https://github.com/fchollet/keras

| Model | (Accuracy %) |
|---|---|
| Kim et. al., [18] | 55.87 |
| Adebayo et. al., [16] | 68.40 |
| Kim et. al., [18] | 67.39 |
| **This paper** | **71.30** |

**Table 4: Evaluation as Textual Entailment task on COLIEE 2014 dataset.**

reduced to two, i.e., similar to a premise and a hypothesis. It is also possible to modify the text from our dataset. Normally, we could join the question text to its corresponding passage text, and regard it as the premise. We could also manually rewrite the answer text where possible by including some phrases from the question text, such that the text reads sensibly. In that case, we can regard the resulting text as the hypothesis. This would make the dataset preparation step similar to the one described in [9]. However, because we do not have the dataset of Biralatei et. al., [9], it is difficult to perform any direct comparison, even though their work is similar to ours in terms of the domain and data. Instead, we utilized the Japanese civil codes dataset which has been released in the context of COLIEE 2014. This dataset has evolved over the years, and an increasing number of researchers are evaluating their work using this dataset.

We encode the input texts following the description given in section 5.1. However, we induce interaction between the input texts at only one level. What this means is that we perform only the intra-sentence attention without any need for the inter-sentence attention. Apart from this modification, every other part of the model remains intact. Table 4 shows the result of our system against three other systems when evaluated on the COLIEE dataset in the context of Textual Entailment. The first and the third are the baseline systems, i.e., the result reported by the authors in [17]. The second is a participant in the COLIEE task [16]. We can see that our model slightly outperforms the reported papers.

### 6.3 Discussion

Table 2 shows the result obtained on the LQA corpus when the main evaluation was done. We see that our model significantly outperforms a TFIDF baseline. Throughout the evaluation, we use the standard accuracy metric. To validate our model, we inspected the questions that were scored correctly by our models but incorrectly by the TFIDF baseline. We give one example of such passage-question-pair. In this particular example, the TFIDF baseline predicted the wrong label for each of the answer options.

**Example** 4:
**Passage**: After being fired, a woman sued her former employer in federal court, alleging that her supervisor had discriminated against her on the basis of her sex. The woman's complaint included a lengthy description of what the supervisor had said and done over the years, quoting his telephone calls and emails to her and her own emails to the supervisor's manager asking for help. The employer moved for summary judgment, alleging that the woman was a pathological liar who had filed the action and included fictitious documents in revenge for having been fired. Because the woman's attorney was at a lengthy out-of-state trial when the summary-judgment motion was filed, he failed to respond to it. The court, therefore, granted the motion in a one-line order and entered final judgment. The woman has appealed.

**Question**:Is the appellate court likely to uphold the trial court's ruling?

- **Answer A (false)**: No, because the complaint's allegations were detailed and specific.
- **Answer B (true)**: No, because the employer moved for summary judgment on the basis that the woman was not credible, creating a factual dispute.
- **Answer C (false)**: Yes, because the woman's failure to respond to the summary-judgment motion means that there was no sworn affidavit to support her allegations and supporting documents.
- **Answer D (false)**: Yes, because the woman's failure to respond to the summary-judgment motion was a default giving sufficient basis to grant the motion.

We can see that predicting a correct answer for this particular example requires the semantic understanding of the underlying text. We conclude that this is evidently lacking in the TFIDF baseline.

Table 3 compares the result of our model with the overall performance of students in 2016 NCBE statistics. We arrive at the percentage score based on the data in Table 1. This is calculated by dividing each score by the total possible score (200) and then multiplying by 100 in order to obtain a percentage score. We can see that our model significantly outperforms the minimum student score. Also, we obtain a better score than the mean student score. We can see that the model shows an appreciable approximation of understanding of the legal technical jargon. We expect to have an improved performance once we have a sizable legal text collection, which we can use to train the Word2Vec algorithm for obtaining the embedding matrix for our vocabulary words. In reality, it is even better if such texts are related to the MBE exam. This will produce semantically rich embeddings that will capture many legal terms. In addition, using extra facts, e.g., as proposed in the second format of the corpus, should improve the performance since many extra details for general learning would be captured.

## 7. CONCLUSION

In this paper, we presented a Legal Question Answering system using a Deep Neural Network technique. Specifically, we employed a LSTM Neural Network which has the ability to retain information much longer than a conventional Recurrent Neural Network. We also described a corpus which has been extracted from the USA MBE exams. We formalize the task as that of Answer-Sentence-Selection, where the system selects the correct answer to a question given a background passage. When compared against a TFIDF baseline, our model displayed a significantly better performance. Similarly, when compared against the human performance based on the statistics available from student performance in MBE Exam. The system obtained a better performance than the mean student score. The proposed task is different from the Textual Entailment task. However, the system shows a good result on a textual entailment dataset. In the future, we would like to obtain more data from Legal tests like the

MBE or any equivalent exams in other countries. We provided a dataset with more information that explains why an answer is correct or otherwise. Intuitively, ML algorithms may learn from the extra information to guide their choice of answer. However, this part is currently lacking in our work. In our future work, we would like to explore how we can improve the performance of our system by incorporating this evidential information as described in section 3. In particular, it would be interesting to compare ML models that take advantage of this information to those who have no access to such information.

## ACKNOWLEDGEMENT

Kolawole J. Adebayo has received funding from the Erasmus Mundus Joint International Doctoral (Ph.D.) programme in Law, Science and Technology. Luigi Di Caro and Guido Boella have received funding from the European Union's H2020 research and innovation programme under the grant agreement No 690974 for the project ”MIREL: MIning and REasoning with Legal texts”. The authors would like to thank the anonymous reviewers who have suggested ways to improve the quality of the paper.

## 8. REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D Manning. Modeling biological processes for reading comprehension. In *EMNLP*, 2014.

[3] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*, 2015.

[4] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.

[5] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer, 2006.

[6] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10, 2011.

[7] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics, 2011.

[8] Laurene V Fausett. *Fundamentals of neural networks.* Prentice-Hall, 1994.

[9] Biralatei Fawei, Adam Z. Wyner, and Jeff Z. Pan. Passing a USA national bar exam - a first experiment. In *Legal Knowledge and Information Systems - JURIX 2015: The Twenty-Eighth Annual Conference, Braga, Portual, December 10-11, 2015*, pages 179–180, 2015.

[10] Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 813–820. IEEE, 2015.

[11] Jianfeng Gao, Li Deng, Michael Gamon, Xiaodong He, and Patrick Pantel. Modeling interestingness with deep neural networks, June 13 2014. US Patent App. 14/304,863.

[12] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701, 2015.

[13] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] Mohit Iyyer, Jordan L Boyd-Graber, Leonardo Max Batista Claudino, Richard Socher, and Hal Daumé III. A neural network for factoid question answering over paragraphs. In *EMNLP*, pages 633–644, 2014.

[16] Adebayo Kolawole John, Luigi Di Caro, Guido Boella, and Cesare Bartolini. An approach to information retrieval and question answering in the legal domain.

[17] Mi-Young Kim, Ying Xu, and Randy Goebel. A convolutional neural network in legal question answering.

[18] Mi-Young Kim, Ying Xu, and Randy Goebel. Legal question answering using ranking svm and syntactic/semantic similarity. In *JSAI International Symposium on Artificial Intelligence*, pages 244–258. Springer, 2014.

[19] Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. pages 0–6.

[20] LR Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 2001.

[21] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[22] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*, 2016.

[23] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014.

[24] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[25] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[26] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

[27] Harry Surden. Machine learning and law. 2014.

[28] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[29] Oanh Thi Tran, Bach Xuan Ngo, Minh Le Nguyen, and Akira Shimazu. Answering legal questions by mining reference information. In *JSAI International Symposium on Artificial Intelligence*, pages 214–229. Springer, 2013.

[30] Peter D Turney, Patrick Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

[31] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[32] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[33] Adam Wyner and Wim Peters. On rule extraction from regulations. In *JURIX*, volume 11, pages 113–122. Citeseer, 2011.

[34] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489, 2016.

[35] Wenpeng Yin, Sebastian Ebert, and Hinrich Schütze. Attention-based convolutional neural network for machine comprehension. *arXiv preprint arXiv:1602.04341*, 2016.