Aston University
Birmingham

Question and answer reasoning on legal text, why and how it can be enhanced

CS4705

MSc DISSERTATION

27/09/2022

NITHIN MATHEW JOSEPH (210199723)
CANDIDATE ID:649014

Abstract:

Unstructured text is difficult to process via language models. Newer trends have shown that with the advancement of neural network models, tasks such as processing unstructured texts have become a walk in the park. The legal domain is a field where unstructured texts are found in plenty along with legal jargons that complicate the sentence structure in a text. Here in this research the author intends to find out a question-and-answer framework that can simplify the process of interpreting such kinds of texts. The benefit of the question-and-answer model is that it helps the user understand the perspective in which the legal text is meant to be interpreted. This research also investigates on named entity recogniser and how it can enhance the question-and-answer framework. The named entity recogniser accomplishes this task by highlighting the entities in the query and the context, to enable the user better ingrain how the query needs to be constructed in order to gain information.

A business aspect of this study will be a product which constitutes the user interface application that is interlinked with the backend models. The paper also highlights the constraints that were encountered when building and synchronising each major component. Depending on the demands and available resources, these limitations could be removed in the future along with the inclusion of new features.

Acknowledgement:

I had like to thank my supervisor Abinaya Sowriraghavan for her exceptional support and guidance in this project. She directed me well whenever I felt stranded at a piece of code or a concept that I was not able to relate. Her words in the meetings encouraged me to better myself. The thought she always conveyed with her words was that I should not complete the project for the sake of completion instead I should try to grasp everything that I ever learned for the project. Her actions were always complementary to the words she preached, which always had a positive effect on me despite my short comings. I was able to learn a lot of things under the tutelage of Abi. I would never be able to repay her consideration whenever I fell one step short of her expectations.

Table of contents:

Contents

Table of figures:

Chapter 1: Introduction

Advancement of technology is something that is inevitable for the human civilization. Information is viewed as the essential resource for securing the future. The way that information is stored has evolved over time. Information used to be recorded on documents in the past, but it is now saved in databases. This is because information is considered more valuable than any form of currency. This transition is observed in technical, medical, political, legal and other domains. This research study focuses on the profession of law where information is judge, jury, and executioner. So, it is vital that the information obtained is not corrupted by any means. The language complexity is a bit harder to swallow than the natural language. Humans with experience in the field still struggle to interpret this language. Artificial intelligence has come a long way from fantasy to reality. Is it possible, to program a computational system to grasp quickly and precisely than a human? The advancement has come so far that it is indeed possible due to the paradigm shift of the human civilisation to the mechanised civilisation.

Search engines that can estimate and provide factual evidence regarding the distance between earth and the sun does exist and it is presently available in the palm of our hand in the form of a compact devices that we hold close to the chest. Which means its only a matter of time before justice becomes obsolete once its automated. This project closely investigates on how neural and deep learning techniques can be utilised to provide an efficient model that can answer the queries posted by an ordinary citizen. In the perspective of law providing knowledge to become one with the judicial system. For the powerless to hold power with the click of a button and for the powerful to become equals with the powerless.

A named entity recogniser and a question-and-answer model were developed as part of this research. These models are fed information through a web application that acts as a user interface. The author explains the limitations, the workaround and the gap that was felt when transforming the theoretical aspects to practicality. Future application of this study is to pave way as a beacon to provide relief for the helpless and power for the ones desperately in need of it.

Chapter 2: Literature Review

The human texts are usually unstructured and even more difficult for humans themselves to understand. So, it can be tricky for a computational hardware to cover so much ground. Over the course of time, humans developed a systematic way to decipher the human texts for machines. The text is broken down to smaller chunks followed by Natural language programming techniques to interpret the text. The condensed phrase NLP refers to natural language programming which literally means programming the language the most natural way for any computational system to understand the human texts.

2.1 NLP Pipeline:

The NLP pipeline is a set of methods to perform analysis, interpretation, experimentation and forecasting on any language. They are compartmentalised into different components to dissect human text and extract key features. The five components of an NLP pipeline is as follows:

1. Language
2. Punctuation
3. Morphology
4. Syntax
5. Semantics

2.1.1 Language identification

NLP has the capability to recognise any language. It needs to have the foresight to identify the natural language from a piece of text provided. If we take for example the google translate when we provide a phrase or a text the first step it initiates is detection of the language. The relative frequency of certain characters stored in the language helps to expose the identity of the language. Let break this down a notch, the number of times the character ‘th’ will be repeated in English language will be enormous compared to any other language. The same concept is the underlying backbone of any language. The occurrence of specific characters can be designated as the signature of a language.

2.1.2 Punctuation

This stage is divided into two mechanisms namely tokenization and sentence splitting. The part tokenization plays in punctuation is to identify the entities present in the text. How does identifying entity help a computational system? This will be a question for which answers can be found in the upcoming pages. Whereas sentence splitting plays the role of understanding the context of the text. The weightage of each sentence can be analysed to extract the key narrative in the case of a passage.

Tokenization: It is splitting of sentence into sequence of characters called words. Usually, it is done based on white spaces. There are corner cases for example like Chile’s, O’Neil, isn’t which needs to be assessed before extraction is executed.

Sentence splitting: In this mechanism the sentences in a text are extracted using key punctuations that mark the end of a sentence. In the English language it is mainly “.”, “!” and “?”. There are special cases that can break the rule laid out for sentence splitting. For example: Mr. Ebin, Alas! I forgot to take the phone., How’s the leg? Thought you had recuperated and so on.

2.1.3 Morphology

Morphology involves breaking down large portion of human texts to smaller nibbles for ease of computation. It is like how a cow grinds the grass it chews for better digestion. This is the core ideology behind the mechanisms mentioned below:

Lemmatization: It is the process of deriving the standardised form of the word. For example, it involves chopping off plural infections like trees to just tree or in the case of continuous tense jumping to just jump. This is achieved by utilising a storable data type to map the change required to the normalized word. This process can be strenuous as it is time and memory consuming, but on the other hand it is high on accuracy. This method is called lemmatization.

Stemming: Like lemmatization in order to obtain the standardised form the method of rules is applied. This involves regular expression which modifies the word of interest.

Ngrams:

An approach used in the morphological methodology is called Ngrams. This method is basically employed to split a sentence based on a delimiter value. That is based on a specific count of words the sentence is partitioned. It resembles an array where each element is stacked very closely. This provides NLP models with adequate information in each iteration so that the model doesn't get overwhelmed.

Unigrams (n=1): They contain series of token with length 1

Bigrams (n=2): They contain series of token with length 2

Trigrams (n=3): They contain series of token with length 3

Figure 1-(Bernhardt, 2021)

2.1.4 Syntax

This stage involves identifying characteristic features in the text and tagging them to their respective labels to define the factors that make up a sentence.

Part-of-speech (POS) tagging: The practise of categorising words found in a text (corpus) based on its definition and context is known as tagging in natural language processing (Part Of Speech Tagging for Beginners, 2022). This involves allocating the token in text which corresponds to part-of-speech (Noun, Verb, Adjective, Pronoun, etc) to their respective tags. Practical models that exhibit this behaviour is Spacy which has the capability of POS tagging.

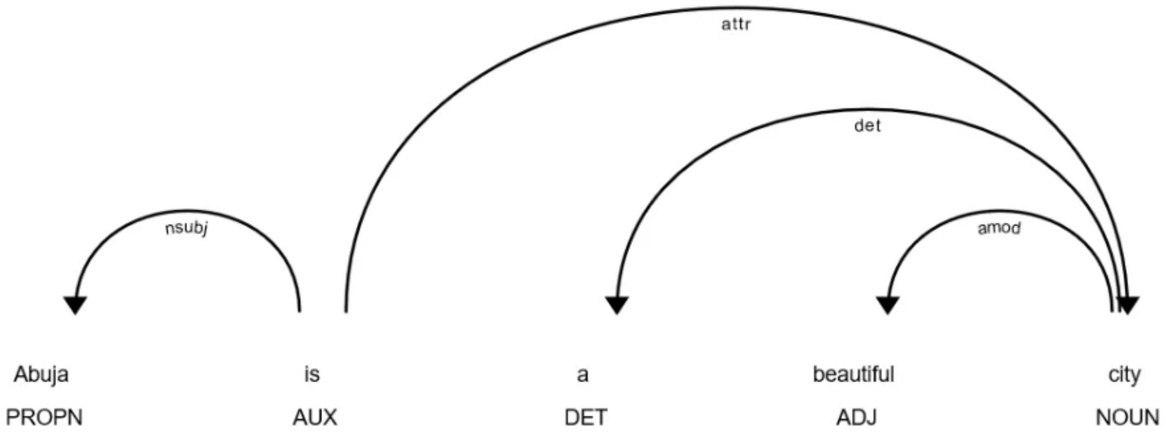

Figure 2-(Part-of-Speech Tagging examples in Python - Jennifer Kwentoh, 2022)

Syntactic parsing: This mechanism is also similar to POS tagging. Here the only varying procedure is assigning the tokens to their respective syntactic role (Subject, Verb, Object, etc). The output of this method will be syntactic labels along with their dependency links.

2.1.5 Semantics

Core of the NLP pipeline lies in the semantics. It constitutes the methods that can be used to identify entities. This makes human speech easier to digest for any computational system. The methods are as follows:

Semantic role labelling: In this method the tokens are marked to their corresponding semantic roles (Agent, Patient, Instrument, Location, etc). The NLP libraries available in python have the capability to assign the tokens from a text to their relative semantic roles.

Entity recognition: In this method the model classifies noun as person, organisation, or location. There are NLP libraries trained on various entity recognition. These packages can be imported in python to assign entities found in texts. This method is the backbone of this research and will be discussed in detail in the succeeding literature review.

Coreference resolution: In this technique the model recognises different words observed in a text refers to the same entity. For example, Modi is the Prime minister of India. As the head of the Indian government, he is morally obligated to act with dignity. Here the words Prime minister and head of the Indian government both refers to the same entity which is Modi.

2.2 Named entity recognition

One of the most recognised data crunching engagements in the field of natural language programming is Named Entity Recognizer (NER). It detects and segregates the key intel observed in the text. The key intel here is referred to as entity and they may be a word or a series of words that refers to the same item (How To Train Custom Named Entity Recognition[NER] Model With SpaCy - NewsCatcher, 2022).

The central function of any entity recognition system has two key steps:

1. Identifying the location of any entities in the text.
2. Labelling entities to named classes (How To Train Custom Named Entity Recognition[NER] Model With SpaCy - NewsCatcher, 2022). For example: George R. R. Martin published his book on “A Song of Ice and Fire” from which HBO adapted the series of Game of Thrones. Here George R. R. Martin will be an entity which will belong

to the class of PER (person) and HBO is another entity which will be labelled to the class of ORG (organization).

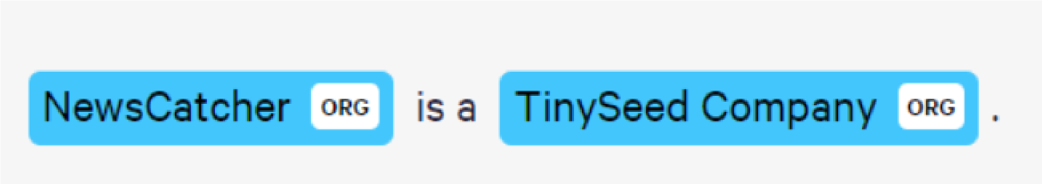

Figure 3-(How To Train Custom Named Entity Recognition[NER] Model With SpaCy - NewsCatcher, 2022)

The NER has the capability to pinpoint the index of the token or sequence of token that corresponds to the entity. This method is named inside-outside-beginning chunking as it helps identify the beginning and ending of the indices of the entities (How To Train Custom Named Entity Recognition[NER] Model With SpaCy - NewsCatcher, 2022).

The next step involves generation of named classes of entities. These named classes depend on the context of NER that needs creation. To give a better understanding of context of NER. Will go over some examples, for medical context of NER the example is as follows "Dr. John prescribed paracetamol for Joyce as he was suffering from fever during his consultation in Queen Elizabeth hospital" Here Dr. John is the entity that will come under the named class of doctor, paracetamol for medicine, fever for disease and Queen Elizabeth hospital for organization. So, the named classes for the medical context of NER will be as follows:

- Doctor
- Organisation
- Patient
- Instrument
- Medicine
- Disease
- Symptoms

In this research, the dataset that is being dealt with is legal. Meaning legal terminologies will be visible in the content of the text. Identifying the entities in it will be complex. So, there are predefined named entity classes that can be used to label these entities. They are as follows:

- Attorney
- Judge
- parties
- Court
- Company
- Case number
- Statue
- Agreement
- Citation
- Jurisdiction
- Location
- Date

These are all role-based entity recognition setup. There are libraries like spacy that have entity recognition system build in from scratch and free to use. They usual work on the principal of machine learning or neural network strategies. There is an inherent ambiguity in plethora of text that has originated from the human civilization (How To Train Custom Named Entity Recognition[NER] Model With SpaCy - NewsCatcher, 2022).

An example would be "The toaster is broken" and "toast to new beginnings" here in the first phrase the toaster means a utility used for toasting the bread whereas in the second phrase toast context corresponds to wishing good fortune. So, ambiguity in the two phrases is visible and human beings have the capability to differentiate them. That's not the case for a computational hardware. A workaround for this would be to utilise more relevant training datasets to teach the model the nuances to differentiate such cases.

2.2.1 Application of NER

- It is helpful in comprehending the high-level perspective, when it comes to large amount of text data.
- Useful when it comes to grasping the context of passage at a glance and separate documents on the basis of their relevance efficiently.
- When it comes to translation application no additional modification of the NER model is required to detect person and location (How To Train Custom Named Entity Recognition[NER] Model With SpaCy - NewsCatcher, 2022).

2.3 Word embedding

Vectors have been found to be beneficial as a way to represent words because they provide a straightforward interpretation and are well suited for usage in various Machine Learning (ML) techniques. The most effective and widely used model for encoding words and sequences of sentences as vectors is the Vector Space Model (VSM), which is typically credited to Salton (1975) and originated in the Information Retrieval (IR) community. In place of using neural networks and embedding layers, new techniques for constructing embeddings have emerged that use word-context matrices to generate vector representations of words (Almeida, F. and Xexéo, G., 2019). According to the distributional hypothesis, word embeddings are dense, distributed, fixed-length word vectors constructed using word co-occurrence data (Turian et al., 2010). When attempting to use analytical techniques on text data, the first issue that arises is probably how to describe it in a form that is accessible to operations like similarity and compositions (Almeida, F. and Xexéo, G., 2019). In the area of information retrieval (IR), one of the first methods to achieve that goal was by using a method to encode each document in a set as a t-dimensional vector, with each element standing for a unique phrase found in that document (Salton et al., 1975). These components might be binary or real values, with the possibility of normalising them using a weighting system like term frequency inverse document frequency (TF-IDF) to take into consideration the varying amounts of information each term provides. With this kind of vector space in place, one can then move on to using these vectors to perform useful tasks like scoring search results, calculating the similarity between document vectors, etc (Almeida, F. and Xexéo, G., 2019).

In this study, this concept is utilised to identify the answer by using the context provided in the query. To put it in simple terms, identify a similarity factor that can pinpoint the answer in the passage.

2.4 Question answering

The main research of the topic is to build a question-and-answer model for the legal text. The research will involve building a model capable of interpreting the query. According to one study, there are three primary processes if the model is created in a question-and-answer (Q&A) format. The steps are as follows: represent the query, represent the document, and check for similarity to determine the importance between the query and the document (Kien et al.,2020).

The NER model discussed earlier is the foundation for the project. NER Models are quite popular when it comes to question answer framework as it helps weed out the answer by detecting the semantic entity of interest. The model can narrow down to a smaller collection of results. For example, "What is the capital of India?" the model will interpret that the expected article should contain the location. This will help the model filter large quantity of data to a short stack of info (Veluri and Z.A.R., 2015).

In this section of the research will be deep diving into the approaches that can be taken for a better question and answer model. The neural network model can be used to recognise and categorise named entities in natural language inquiries. The paper on "Named Entity Recognition and Question Answering Using Word Vectors and Clustering" (Veluri and Z.A.R., 2015), it is hypothesised that the prelabelled questions in training corpus will be beneficial for the NER system. The researchers on this paper did a comparison study to determine the performance of their model when the train data contained annotated questions against free text corpus. What they observed was the model that was trained with annotated queries provided better F-measure. This convinces the statement that including questions in the training corpus will benefit in the long run for a question answer system. During their investigation they learned that a mixture of annotated and free text questions enhanced the model's performance (Veluri and Z.A.R., 2015).

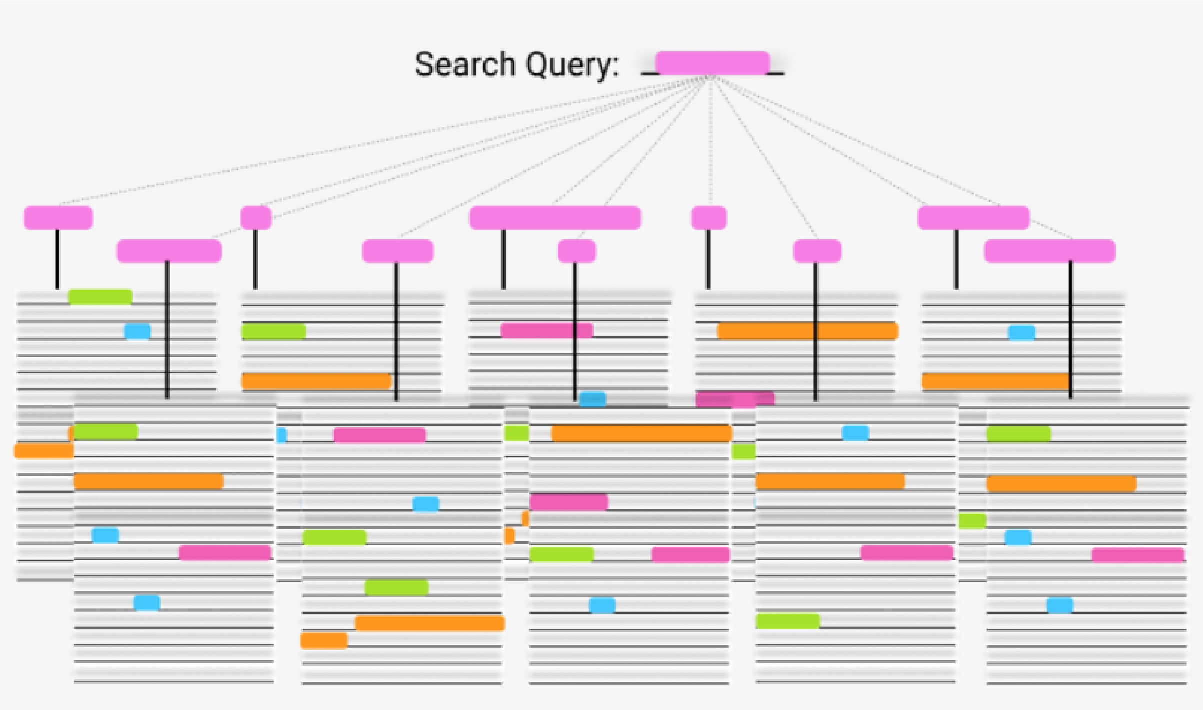

Figure 4-(How To Train Custom Named Entity Recognition[NER] Model With SpaCy - NewsCatcher, 2022)

2.5 Three neural networks of interest for legal corpus

The figure below provides the legend for the components constituting the neural networks mentioned below:

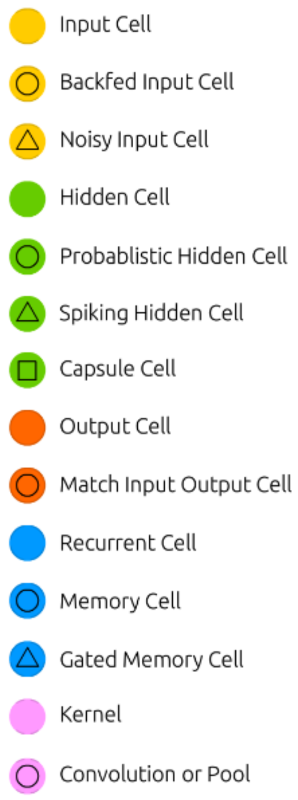

Figure 5-(VEEN, 2022)

2.5.1 Convolutional neural network

A convnet is a sequence of layers that are sequentially applied to the input. Layers are applied in triplets that consist of convolutional layer, a non-linearity layer and a pooling layer. After a succession of these triplets the output is sent via one or more fully connected layers with their matching non-linearities. Convolution is a special form of linear mapping from M to N dimensions. Therefore, we can write convolution as a matrix multiplication. The matrix has a very restricted form with lots of zeros and repeated entries. The matrix form offers little information. The natural extension of convolution can be to two, three, or more dimensions. The specific formula is a bit complicated, but the idea is fairly straightforward. Convolutional neural network (CNN) uses a convolution layer as one of its primary components. As the neural network is trained, the kernel filters are gradually configured. Each kernel's size is smaller than the input's size. Each kernel filter makes use of the input to produce an activation map. For convolution, the filter moves over the image's height and breadth and then executes the dot product of each filter element. Based on the element of the input image, this process is repeated numerous times to produce the activation map.

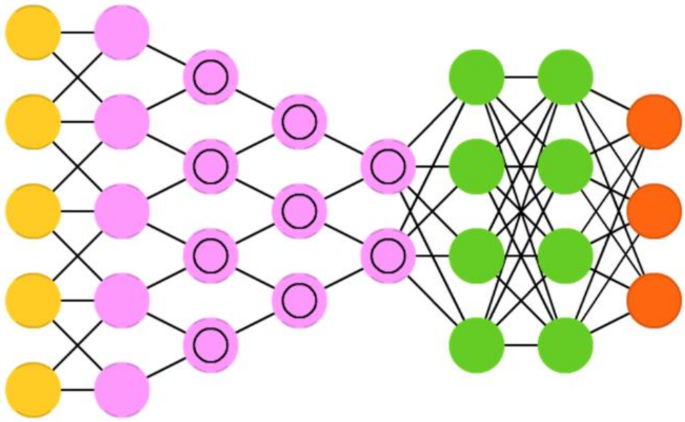

Figure 6-(VEEN, 2022)

The non-linearity is observed via the activation function. Rectified linear unit (RELU) is most used activation function. It traces the output of the activation map to the highest possible value and if the element in the activation map is negative, the function traces it to zero. Further transformation is achieved by applying pooling. By combining the output from several small regions into a single output, pooling decreases the dimensionality of the feature map.

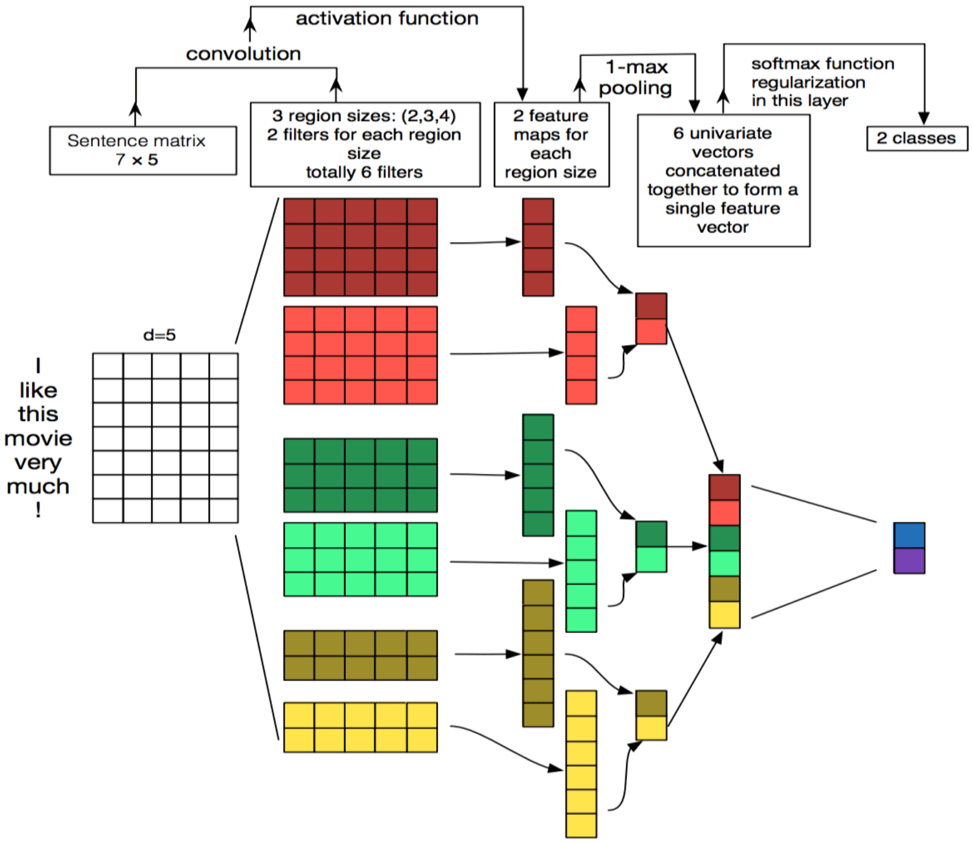

Figure 7-(Brownlee, 2022)

Convolution takes advantage of three concepts to enhance machine learning systems.

Sparse interactions: Each output node in completely connected layers is functionally reliant on every input. Each output node in a convolution layer is dependent on a limited number of input nodes.

Parameter sharing: The weights used in an output node's computation are shared by all output nodes, which causes computation to be redundant.

Translational equivariance: If we translate the input. The output is the same as earlier, but it is only translated by the same amount. Reducing the set of parameters to be trained.

2.5.1.1. Challenges with CNN model

Another research that caught the interest while investigating convolutional neural network model was the research paper on “Answering Legal Questions by Learning Neural Attentive Text Representation” (Kien, P.M. et al., 2020). In this study, convolutional neural networks (CNNs) and attention mechanisms were used to extract pertinent data and match similarities found in the question and answers. Sentence and paragraph encoders that extract crucial elements from any texts are made using the CNN. The researchers of the above-mentioned paper implemented a negative sampling paradigm to train the CNN model. The process classifies the articles that are pertinent to the query as positive, while the articles that are irrelevant to the query as negative. The model's performance was assessed in comparison to models from the most recent ten years. It outperformed Bert's rival, Birch (Yilmaz, Z.A. et al., 2019), in the legal realm by a wide margin. It was nevertheless unable to surpass a Birch variant that was trained on article titles. To comprehend the significance of each stage, the researchers conducted an ablation study. They were able to understand how the performance of their model as a whole was affected by each stage. According to the findings of the study, this strategy appears to have been successful (Kien, P.M. et al., 2020).

But there is a key issue, for this type of model. That is, there is no accepted approach to assess how well the NLP model performs on legal text. The method discussed for evaluation in the paper on “Answering Legal Questions by Learning Neural Attentive Text Representation” (Kien, P.M. et al., 2020) were recall and normalised discounted cumulative gain (NDCG) which compared the performance of the model that combined sentence encoder and paragraph encoder on the foundation of attention mechanism against existing models. Recall is calculated as the sum of all true positives minus all false negatives. A document's location in the result list is used by NDCG to determine if it is of any gain. Gains are achieved from the top of the result list to the bottom, with lower rankings discounting each output's gains (Järvelin, K. and Kekäläinen, J., 2002).

The availability of train data is an additional issue that affects models like these. In the above-mentioned research study, the researchers made use of two constructed datasets. The constructed datasets contained Vietnamese legal papers in the format of question-and-answers. This question-and-answers datasets constituted numerous legal questions with relevant articles for each question. The raw legal papers were taken from official sources, while legal advice websites provided the questions. Title and content were included in the query. The researcher opted to preserve good topics, change uninformative titles, and eliminate some portions of the text because it was too long and confusing. The legal corpus had many iterations of the laws and regulations, which were eliminated to eliminate duplication. This was accomplished with the aid of legal professionals. Because the data set was of poor quality, it needed to be altered and purged of unimportant elements in order to perform better. Even though it’s a workaround, utilizing a CNN model doesn’t negate the interference caused by these issues (Kien, P.M. et al., 2020).

2.5.2 Recurrent neural network

Convnets is an architecture designed specifically for managing data grids (example: images). Which by using sparsity and parameter sharing, translation invariance is overcome.

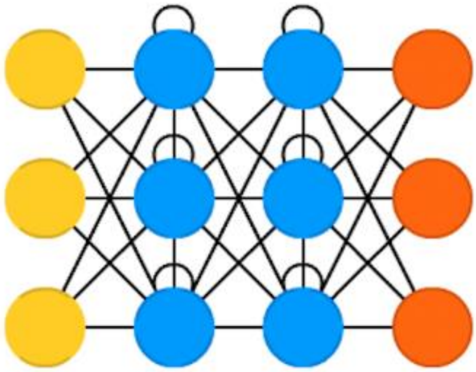

Figure 8-(VEEN, 2022)

An architecture that is specifically designed to handle sequences of data is the recurrent neural network (RNN). Sequence examples include:

- Time series data (market rates, GDP, weather)
- Audio signals
- Language data (sequences of characters, words, etc)

Variable number of elements, past events influence the present, and future events influence the present.

Handling temporal structure: using a feedforward network to simulate the mapping between a variable of interest and a window of previous observations.

z_t= f $(x_t, x_{t-1}, \ldots, x_{t-p})$

This is fundamentally a p-size kernel, 1-dimensional convolution method. This is referred to as brute force approach where the model explicitly depends on the past dataset. For dynamic systems the approach will be different. The state s_t encompasses all facets of the world under consideration in a dynamic system (example: weather, market prices, …). Assume that the state is fully dependent on all prior observations without making any further assumptions.

s_t= f $(x_1, x_2, \ldots, x_t)$

This becomes intractable as the t value grows. Dynamical systems commonly operate on the Markov assumption, which states that the current state contains all information related to previous observations. A much more manageable formulation is made possible by this assumption.

s_t= f (s_{t-1}, x_t)

To put it bluntly the concept is not interested in the entire state, specifically some output. An example of this would be identifying the action occurring in a video clip. The output is assumed to be a function of state s_t at time t using alternative compact representations of the state.

z_t=g (s_t) = g (f (s_{t-1}, x_t))

RNNs are a specific kind of dynamical system, where h_t stands for the state that is stored in the activations of the neuron at time t. h_t and h_{t-1} are functionally dependent on one another. The ability to accept feedback connections distinguishes this network from conventional feed forward networks. Purely feed-forward networks lack internal dynamics. No matter the time, such a network always generates the same output, z_t. In RNNs, feedback connections lead to persistence of input activations that resemble memory. RNNs feature internal dynamics that lead the internal state to change every time the same observation is seen, resulting in a different output.

2.5.2.1 Challenges with recurrent neural networks

Although the recurrent network is a straightforward and effective model, training it is challenging in reality (Pascanu, R. et al., 2013). The difficulties of vanishing gradient and exploding gradient are among the key causes of this model's ungainliness (Bengio, Y. et al.,1994). The term "exploding gradient" describes the significant rise in the gradient's norm during training. These occurrences result from the long-term components, which can radically generate tenfold more than the short-term ones, leading to overdrive. The converse tendency, known as the vanishing gradient, prevents the model from interpreting connection between temporally far-off events while at the same time the long-term components move exponentially quickly to norm 0 (Pascanu, R. et al., 2013). To explain it in simple terms recurrent networks must learn which previous inputs must be saved to generate the required output at that moment. In order to create a sufficient input storage with gradient-based learning techniques, the current error signal must travel back through the feedback links to earlier inputs. However, when the short time delay between inputs and respective teacher signals are prolonged, conventional backpropagation suffers from an excessively long learning period. The error signal tends to vanish due to this excessively long learning time (Hochreiter, S., 1998).

2.5.3 Transformer

The Transformer neural network has quickly overtaken competing neural models like convolutional and recurrent neural networks as the industry standard for natural language processing, outperforming them in applications requiring both natural language interpretation and natural language synthesis (Vaswani, A. et al., 2017). Model pretraining enables models to be trained on general corpora which makes them receptive to objectives (McCann, B. et al., 2017). Systems are needed to train, evaluate, scale, and modify the Transformer model on many platforms due to its widespread application (Thomas, 2020). Transformers' architecture is influenced by Google Research's ground-breaking tensor2tensor package (Vaswani, A. et al., 2018) and the original BERT source code (Jacob et al., 2018). Transformers was made to replicate the common NLP machine learning model pipeline, which includes processing input, using a model, and making predictions (Wolf, T., 2020).

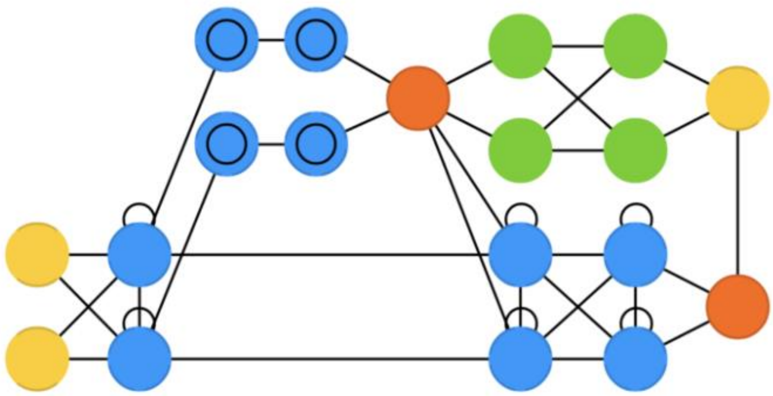

Figure 9-(VEEN, 2022)

This neural network known as a transformer learns context and monitors relationships in sequential input. Transformer models use an expanding collection of mathematical approaches known as attention or self-attention to find minute relationships between even far-flung data pieces in a series. Transformer models are essentially huge data processing encoder/decoder components. Positional encoders are used by transformers to tag data items entering and leaving the network. These tags are followed by attention units, which calculate a sort of algebraic map showing how each element connects to the others (Merritt, 2022).

Practically, all models adhere to the same hierarchy of abstraction: a base class implements the processing graph of the model, starting with an encoding, moving through a sequence of self-attention layers, and ending with the encoder hidden states. Each model's base class is unique and closely resembles the model's initial implementation, giving users the freedom to understand the inner workings of each distinct design. To facilitate easy expansion, each model is typically built in a single file. Then there is the tokenizer class, which is derived from a single base class, that can either be manually customised or created from a corresponding pretrained model. These classes manage the encoding and decoding of input sequences in accordance with a model's unique tokenization procedure. They also maintain the vocabulary token-to-index map for their respective model. Python-based tokenization is frequently too slow for training on very big datasets. Each Transformer can be coupled with one of a number of already-implemented heads with outputs suitable for typical tasks. These heads are built as extra wrapper classes on top of the base class, enhancing the Transformer's contextual embeddings with a particular output layer and an optional loss function (Wolf, T., 2020).

In what is referred to be multi headed attention, attention questions are processed in parallel by computing a matrix of equations. These resources enable computers to recognise the same patterns that people do (Merritt, 2022).

To sum up, because Transformer and pretraining are more significant in NLP, it is crucial that academics and end users have access to these models. An open-source framework and community called Transformers was created to make it easier for users to acquire large-scale pretrained models to develop, explore and execute them in downstream tasks with cutting-edge performance. Since its debut, Transformers has had substantial organic growth and is positioned to provide essential infrastructure while easing access to new models (Wolf, T., 2020). The Transformer architecture is used by the Spacy model, on which the current project is based.

2.5.3.1. Challenges with Transformer model

It has been demonstrated that the present generation of pre-trained language production models of transformers (Wolf, T. et al., 2019) are reasonably good at detecting syntactic cues like noun attributes, third - person pronouns, prepositions, or co-referents as well as semantic cues like entities and relations. However, they have not performed well at detecting various viewpoints, the global context, or relational extractions (Braşoveanu, A.M. and Andonie, R., 2020). This may be because of biases that may already be present in the embeddings and that may lead to more spreading in the downstream activities (Gonen, H. and Goldberg, Y., 2019).

One of the key features Transformers is lacking is better visualization methods. That is to say, generally the framework should focus on the entire Transformer models' lifespan, which includes their input data sources, attention maps, and outputs. Ultimately, mistakes found when building such models can originate from a variety of places, including the text corpora, an arbitrary network layer, or even an external Knowledge Graph that could provide some data for the model. Without visualization aids, tracking such mistakes would be quite time consuming and redundant grunt work (Braşoveanu, A.M. and Andonie, R., 2020).

Chapter 3: Methodology

3.1 Legal dataset

Legal documents can be lengthy and intricate. Latent features abound in legal writings. Without a particular level of knowledge in that field, it is challenging to understand or even translate the legalese. There are five main categories of legal texts: contracts, statutory, administrative and case law. In the dataset that was obtained some of the datapoints constituted at least one of the main categories. Different sorts of laws have distinctive properties that make some computational techniques more relevant to accomplish specific tasks. Collecting all the unique corner cases in this type of dataset is like finding a needle in the haystack. The more redundant the format of the legal article is, the more obvious the content becomes and thereby making it easy for automating it to a formal representation. Every type of law is an own collection of conditions. Public rules and regulations conform to a predictable pattern which have a reasonable organised composition. Nevertheless, because public law does not aim to address every eventuality and situation that can arise, judges frequently have to interpret statutes. Private contracts, on the other hand, are not connected to this web and encompass a wide range of potential outcomes suitable to the relationship being formalised. This means that contracts are more likely than public law to emerge from the chaotic, ambiguous realm of natural language, and it's possible that the emergence of distributed ledgers and related information technologies is hastening this transition. Laws other than private contracts are also curated in a way that is advantageous (Nay, J., 2018).

Legal datasets are incredibly difficult to get access. Most of them may have confidentiality in the textual content provided in it. In the initial stages to build the NER model a scrapping tool was built from scratch to extract data from html pages that contained legal articles. The category of the legal articles was from England and Wales family county court. Due to privacy restriction this dataset had to be dropped. So, after further research obtained legal dataset based in India which had no such constraints. The authors of this dataset were kind enough to provide the pre-processed format along with explanation on how they obtained it. Which is pretty much similar to the initial attempt that was carried out for the British legal dataset. The only difference being they had a tool which collated the hyperlinks of interest for the legal articles followed by the scrapping tool which utilised the links for extraction.

3.2 Ethical Implications

The dataset is where the root of ethical implications originates from. If the information that has been acquired for training a model is sensitive. Then it needs to follow the right procedures so that data points can be held accountable. The results showcased in that very project can have exemplary value only if the datapoints have accountability. On the other hand, when the benefit of the cause outweighs the harm, then the implications are admissible.

Understanding the meaning of private data protection: “one in which the publicity of court trials as a rule of law guarantee trumps the individual’s wish to hide oneself from others in public space” (Langford, I., 2009). Although this regulation is outdated, it nevertheless adheres to the idea that any information that cannot be shared without the subject's permission is considered confidential. There is a contradicting revelation to that law wherein convicts or terrorists have right to privacy. The contradictory implication of this example is that, first, the public needs to be made aware of such criminal trials so that they can keep an eye out, and second, the effects of the convict's or terrorist's action need to be made clear to the public to decrease the likelihood of bringing up such personal in the society.

Therefore, due to the high benchmark for privacy protection, research on legal NLP is evaluated brutally when it comes to ethical implications. This is not to argue that publicly accessible data cannot be regarded as accessible even if it includes some but not all confidential data. Like other researchers, the primary moral obligation of legal NLP researchers is to the impartial pursuit of truth as they perceive it, not to substantive purposes which are extrinsic to that quest (Resnik, J., 2011).

3.3 Tech Stack

This research involved many technologies. The majority of the code in this research paper is built on the language called python. Python is a line-to-line interpreter. It is a language that is currently trending among engineers. It is polished and well-built unlike its predecessor perl. To better explain the Pros of python the cons of its legacy language can be looked at. The issue with perl is that it never errors out the issues that might be lying in the code. This is exhausting in the perspective of a programmer as the individual will have a hard time debugging the code whereas python specifically errors in detail each issue that it has identified in an ordered fashion. This reduces the effort required when it comes to debugging. The pro of perl is that its seamless sync with the Linux environment. It is able to execute the syntax available in the Linux environment via the tick command without any installation or plugin specific to operating system.

The spacy module is something that has been deployed in this project for NER model creation. It is a custom model that can be utilised to build new language classes from scratch. It is easy to construct with a few lines of code and prior knowledge on the NLP pipeline.

A community called Hugging face has prebuild modules that have the capability for modifying datasets to specific formats and models such as Bert in their libraries which can be programmed to build variations of the Bert depending on the dataset its trained on. In this research paper the code on question-and-answer model is built on one of the hugging face communities' libraries. The datasets are also transformed using one of their modules as well.

Flask is also another tech which has been researched on for this paper. It is the final component that is used to gel the models together. Flask is a python package that serves as a web framework to create web apps. Design apps without having to be concerned with low-level aspects like syntax, multi thread supervision, and other issues (What is Flask Python - Python Tutorial, 2022). Flask is a module that is built for python. This was created by Armin Ronacher, who served as the team leader of Poocco, a worldwide group of Python aficionados. The Werkzeg WSGI toolkit and the Jinja2 template engine serve as the foundation for Flask (What is Flask Python - Python Tutorial, 2022). In this research Werkzeg WSGI toolkit is mainly used, as it helps to obtain request and response from the server that is being hosted by flask. As all the models that needs to be synced with flask are deployed in google colab. Flask was also instantiated in colab. It is known that flask cannot host a local server from colab. Hence the module flask_ngrok is used that helps mitigate this issue. This command diverts flask to host in a virtual server.

3.4 NER training dataset annotation

Standard building block of any NER model is the annotation section. The annotated dataset is what helps the model clearly distinguish entities in the test dataset. There are tools that help in annotating the dataset. The one that was used in this research generates an annotated dataset in json file format which contains the sentence of interest, starting index, ending index and tag corresponding to the entity class. The explanation as to how it is used is as follows. While training the model the entity in the sentence is recognised via the start and end index provided in the json file. The class of the entity is memorized via the tag. Once the learning stage is completed the model is provided with a test dataset. Where it is evaluated on its accuracy. The annotation must be done for the entire training dataset which is physically taxing.

Figure 10-(NER Annotator for SpaCy, 2022)

The dataset that is covered in this study is from a legal background. Hence the tags that needs to be assigned need to be linked with legal terminologies. This is implemented using the annotation tool that is accessible via the website (NER Annotator for SpaCy, 2022).

3.5 NER partial annotated training dataset

For this research investigated means to annotate without tedious grunt work. This is done by employing a dataset that has only been partially annotated, with only a small portion of named entities labelled and all remaining tokens by default classified as non-entities. The default classified entities are given the value of zero whereas the entities are provided proper labels. The process here is to identify the true negatives and false negatives from the non-entities. As the non-entities clearly has no critical role in the construction of the model. The negative data points play the crucial role of helping the model identify a pattern. So, an iterative method guided by constraints can identify false negatives in a noisy collection and de-weight them to provide a weighted training set. This model will need to pick up skills. Wherein each token is given a weight using an iterative binary classifier indicating the chances of it being correctly identified. In this stage the method teaches how to obtain relevant datapoints for the entities in the dataset so that it can iteratively increase the chances of finding the next set of valid datapoints. This concept is called constrained binary learning (CBL). To put it in Layman terms an algorithm that iteratively recognises true negatives for the NER model while at the same time learns (Mayhew, S. et al., 2019).

It would involve assigning all negative aspects instance weights, thereby giving false negatives low weights and all other instances high weights. This is done so that the system is able to recognise the false negatives and obtain a pattern for recognising them in the succeeding datapoints. The binary classifier trained on partially annotated dataset helps to estimate the weights for each element. To illustrate further in the partial dataset in which not all but some are indicated as positives and no negatives marked. It is assumed that the ones that are not positive to be negative instead of them being unlabelled. This creates the problem of the dataset being noisy due to positives being marked as negatives. Hence training on this dataset will lead to a noisy classifier (Mayhew, S. et al., 2019).

There are two ways to go about it:

1. Find false negatives and properly identify them.
2. Find and eliminate false negatives (Mayhew, S. et al., 2019).

The former could result in noise addition that is worse than the initial partially annotated dataset. The asymmetry in the penalties makes the latter more forgiving. It is crucial to eliminate all false negatives from negatives, but doing so is usually not problematic, particularly in NER where negative instances predominate. For better performance instead of eliminating false negatives the method of instance weighting is implemented, as each instance is assigned a weight based on the conviction the mark up of the instance generates. A weight of 0 indicates that the model won't be updated despite the loss this instance experiences during training (Mayhew, S. et al., 2019).

P: Positive tokens

N: Noisy negative tokens

C: Constraints

T = N ∪ P
V <- Initialize T with weights (optional)
While stopping condition not met do
L <- Train (T, V)
$\bar{T}$ <- Predict (L, T)
T, V <- Inference ($\bar{T}$, C)
End while
Return L (Chang et al., 2007).

The while loop, which repeatedly iterates over training on T, predicting on T, and correcting those predictions, houses the algorithm's main logic. Here T corresponds to the union of positive token and the noisy negative tokens. During the training stage the instances are provided a weight of V which in the initial stage is set to one or to a specific value based on prior information. Followed by execution of prediction on T. In the inference stage the predictions along with the constraints is used to tag the new labels on T and set up new weight of V (Chang et al., 2019).

To give weights to instances in the second phase of CBL, we use the training data from the first phase. That is:

$$v_i = \begin{cases} 1.0 & if\ x_i \in P \\ P_L(y_i = 0|x_i) & if\ x_i \in N \end{cases} \qquad \text{(Chang et al., 2007)}$$

"Where $P_L(y_i = 0|x_i)$ is understood as the classifier's confidence that instance x_i takes negative label. If the classifier has accurately learned to detect entities, then for all the false negatives in negatives, $P_L(y_i = 0|x_i)$ is small, which is the goal" (Mayhew, S. et al., 2019). To learn a model, we ultimately pass the original multiclass partially annotated dataset and final weights V to a typical weighted NER classifier (Chang et al., 2007).

This approach was initially attempted but due to the complexity in generating the required partially annotated dataset dropped out on the experiment.

3.6 Bert based Question answering model

Hugging Face community, provides prebuilt modules for programming models like Bert to construct different iterations of it. These iterations are built based on the dataset on which the model gets trained on. In this research a variation of Bert is constructed to build the question-and-answer model. The aim of a question answer model, given a question and a paragraph, focuses on pinpointing the precise location within the paragraph that provides the answer for the question (How to Explain HuggingFace BERT for Question Answering NLP Models with TF 2.0, 2022). It is tempting to utilise attention weights to describe the behaviour of BERT because it is an attention-based model. Attention weights represent the inputs that are significant to a certain output assignment (Bahdanau, D. et al., 2014). BERT (base) includes 12 layers with 12 attention heads (How to Explain HuggingFace BERT for Question Answering NLP Models with TF 2.0, 2022), whereas a regular seq2seq model normally has one attention mechanism (Bahdanau, D. et al., 2014) that indicates which input tokens are of concern. Here the concept of attention mechanism refers to the mathematical equation that has the capability to find a link between completely unique pieces of data in a sequence (Merritt, 2022). Adding to this, BERT layers are linked which makes them the centre of focus rather than the words (How to Explain HuggingFace BERT for Question Answering NLP Models with TF 2.0, 2022). In order to focus on various patterns, attention heads pay close attention to the direct object of verbs, determiners of nouns and objects of preposition (Clark, K. et al., 2019). Each attention pattern is obfuscated to allow BERT's extensive language modelling (How to Explain HuggingFace BERT for Question Answering NLP Models with TF 2.0, 2022). Studies have shown that attention weights cannot be immediately interpreted (Gino et al., 2019) and that they can be deceptive as explanations in general (Sarthak and Byron, 2019). However, this does not mean attention weights cannot be effectively used for testing models (How to Explain HuggingFace BERT for Question Answering NLP Models with TF 2.0, 2022). They are useful for scientific probing activities that aid in understanding the behaviour of models, even if they are not as useful as tools for end user interpretation (Clark, K. et al., 2019). Gradients in a trained deep neural network can be used to infer the relationship between inputs and outputs. Gradients calculate how much a change in each input dimension would alter predictions in a constrained area surrounding the input (How to Explain HuggingFace BERT for Question Answering NLP Models with TF 2.0, 2022). The research suggests that basic gradient explanations are more accurate and reliable than complex ones, despite the simplicity of this strategy (Adebayo, J. et al., 2018).

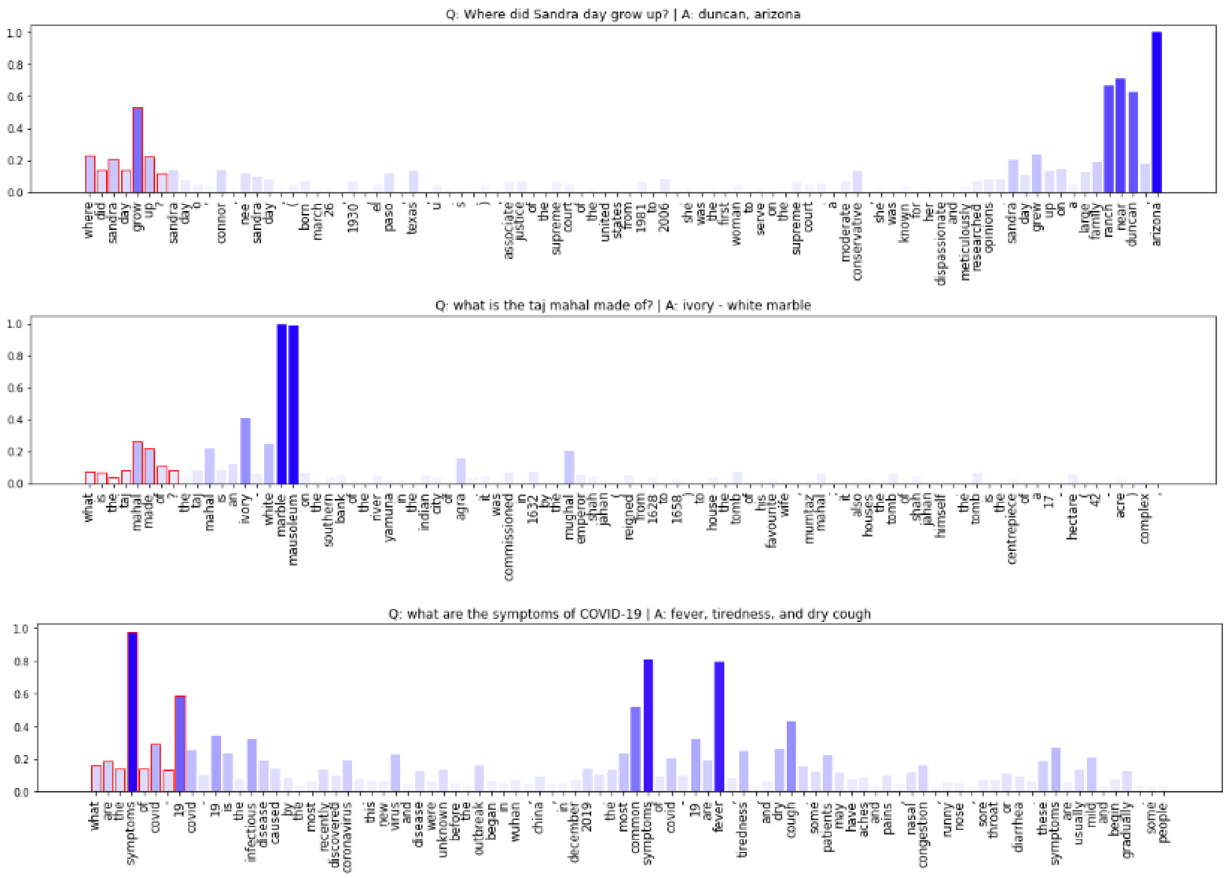

Figure 11-(How to Explain HuggingFace BERT for Question Answering NLP Models with TF 2.0, 2022)

To explain this concept the attribute of TensorFlow 2.0 called GradientTape is mobilized. To record operations on a group of variables GradientTape is automatically differentiated and the model's outcome for a specific input can be explained by (How to Explain HuggingFace BERT for Question Answering NLP Models with TF 2.0, 2022):

- Launching GradientTape and observing the input variable.
- Calculating the model's forward pass.
- Obtaining gradients between the monitored input and the desired output.
- Interpreting justification from the normalised gradient. (How to Explain HuggingFace BERT for Question Answering NLP Models with TF 2.0, 2022)

A similar model is utilised in this study to generate a framework sturdy enough to interpret answers for the legal queries and legal context provided as inputs.

3.7 GUI Prototype framework

This study incorporates a simple user interface to display the query, the context and the answer, to enable non-technical experts to easily observe the Q&A model without much prerequisite knowledge. Modules like tkinter, PyQt5, PySimpleGUI, Kivy and many others can be deployed to visualize an interactive framework between the user and the system. The module that was used in this study was the tkinter. It is flexible and compact which gives the experience of a desktop application. It is easy to program using object-oriented class framework. It provides features like dialog box, scroll bar, radio buttons, browser button and many more which can be inserted on the grid setup via the module. The latest python versions have this module inbuilt which reduces the hassle of pip installing the module via console.

Initial it was planned to use tkinter to build the user interface. Its simplicity and ease of built was not challenging enough. The approach was replaced by flask to provide a better user experience.

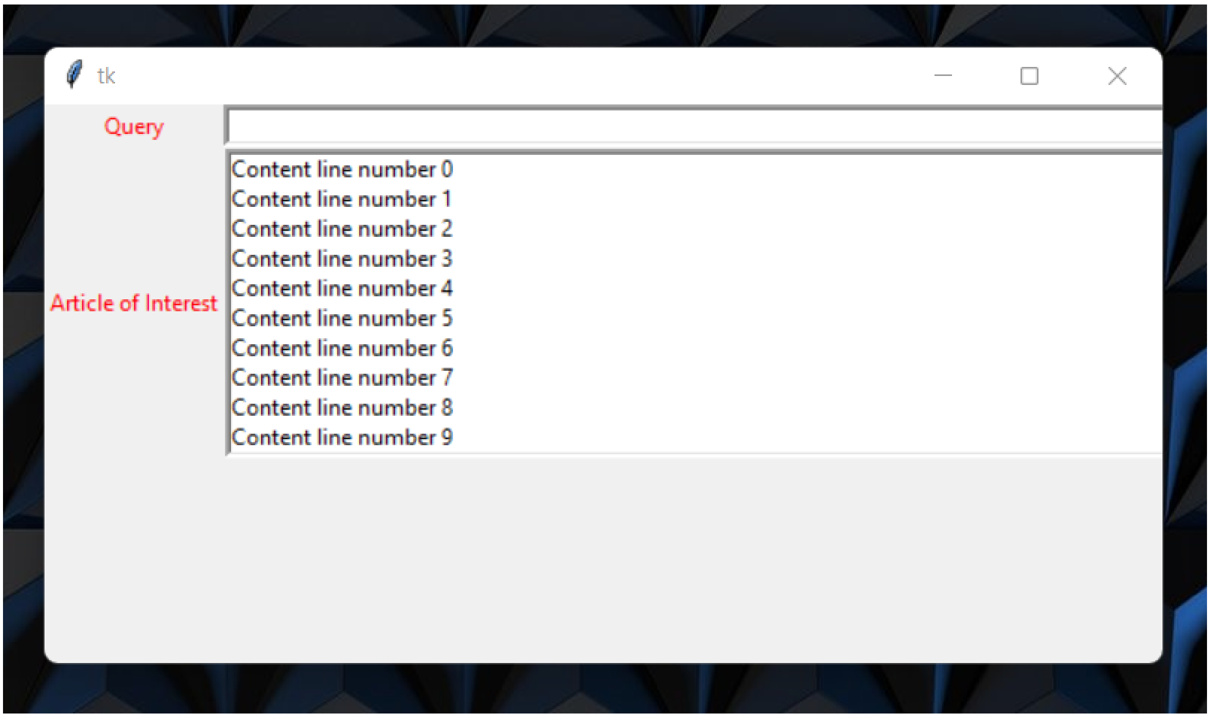

Figure 12-Prototype GUI

Chapter 4: Project management

The below table is the project plan that was proposed in the dissertation proposal

Task	May	Jun	Jul	Aug
Background research				
Framework initial structure				
Data Collection				
Data Analysis				
Implementing the framework				
Simulation of framework				
Identify the issues				
Debug the issues				
Modify the framework				
Run test data				
Evaluate the performance				
Correlation analysis against recent models				
Report				

The Trello board provides a brief glimpse on the entire routine that was followed for this research.

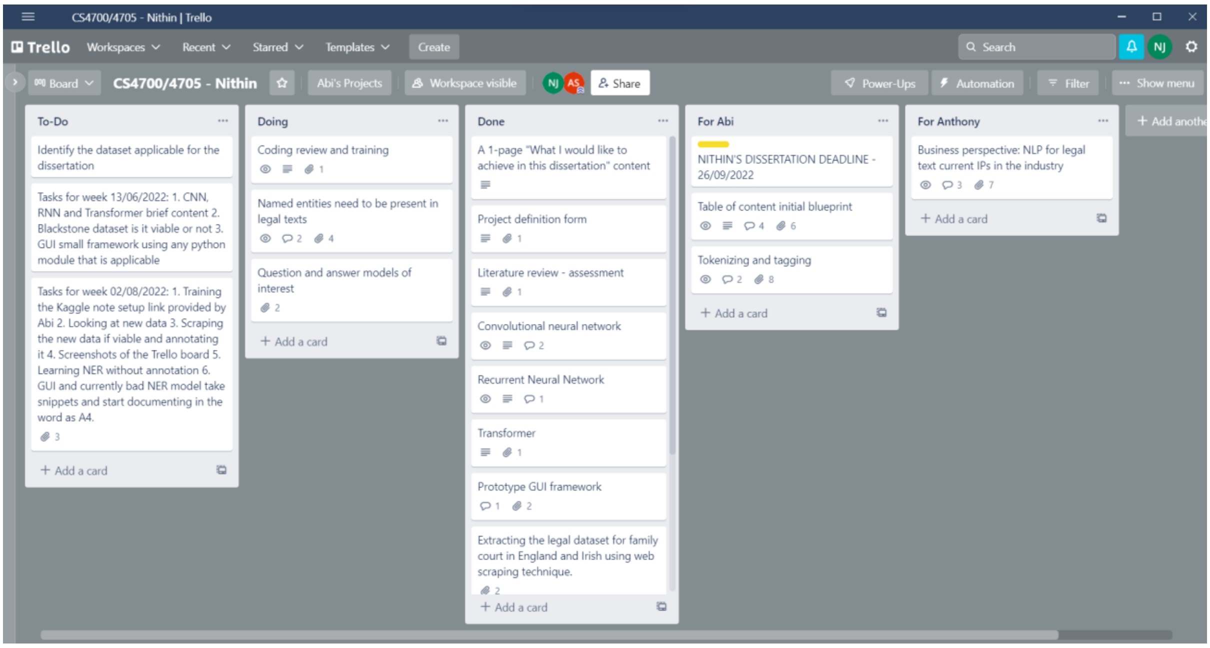

Figure 13-Trello Board dashboard part 1

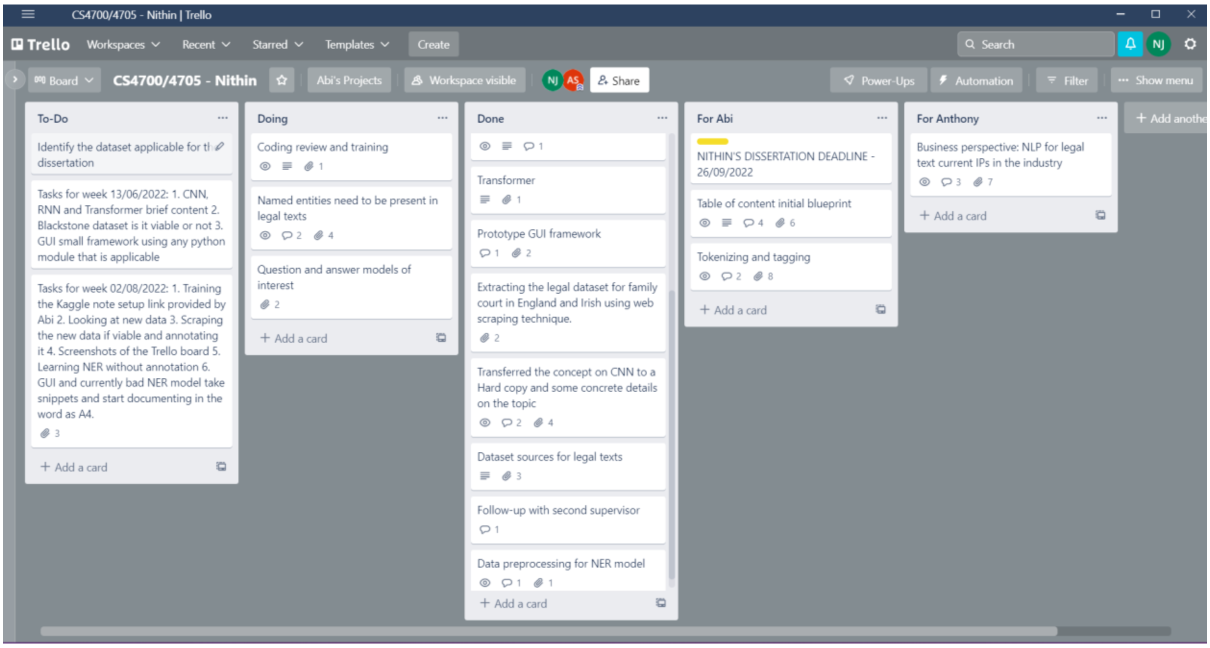

Figure 14-Trello Board dashboard part 2

4.1 Background research

In this study the initial research was on the dataset that can be utilised to build the NER model. The constraint was that the dataset had to be legal. When skimming through different research papers, the initial one that caught the eye was the research on Blackstone model. The Incorporated Council of Law Reporting for England and Wales research division (ICLR&D) created an NER model using authorised legal dataset (Case Genie goes live!, 2022). Unfortunately, this dataset was not provided in the repository.

Another source was a blog that provided the link to a html page. This html page contained UK government legal dataset (BAILII - Reproduction and Copyright, 2022). To extract the info from the html page the module called beautiful soup was utilised. The module is a web scraping method that enables the user to utilise efficient programming to handle intensive work (Paruchuri, 2022). As work progressed it was observed that the dataset extracted using web scrapping could no longer be used due to its copyright policy. It is prohibited to preserve the search results or store the HTML page in any form (BAILII - Reproduction and Copyright, 2022). Hence the development of the code for extraction of dataset from html page was scraped.

4.2 Data collection

After further investigation obtained Indian legal dataset via github repository. The researcher's had mentioned in the repository that they would be willing to provide the datasets, provided their names get mentioned in this research study (Malik, V. et al., 2021). This concluded the data collection process stated in the project plan.

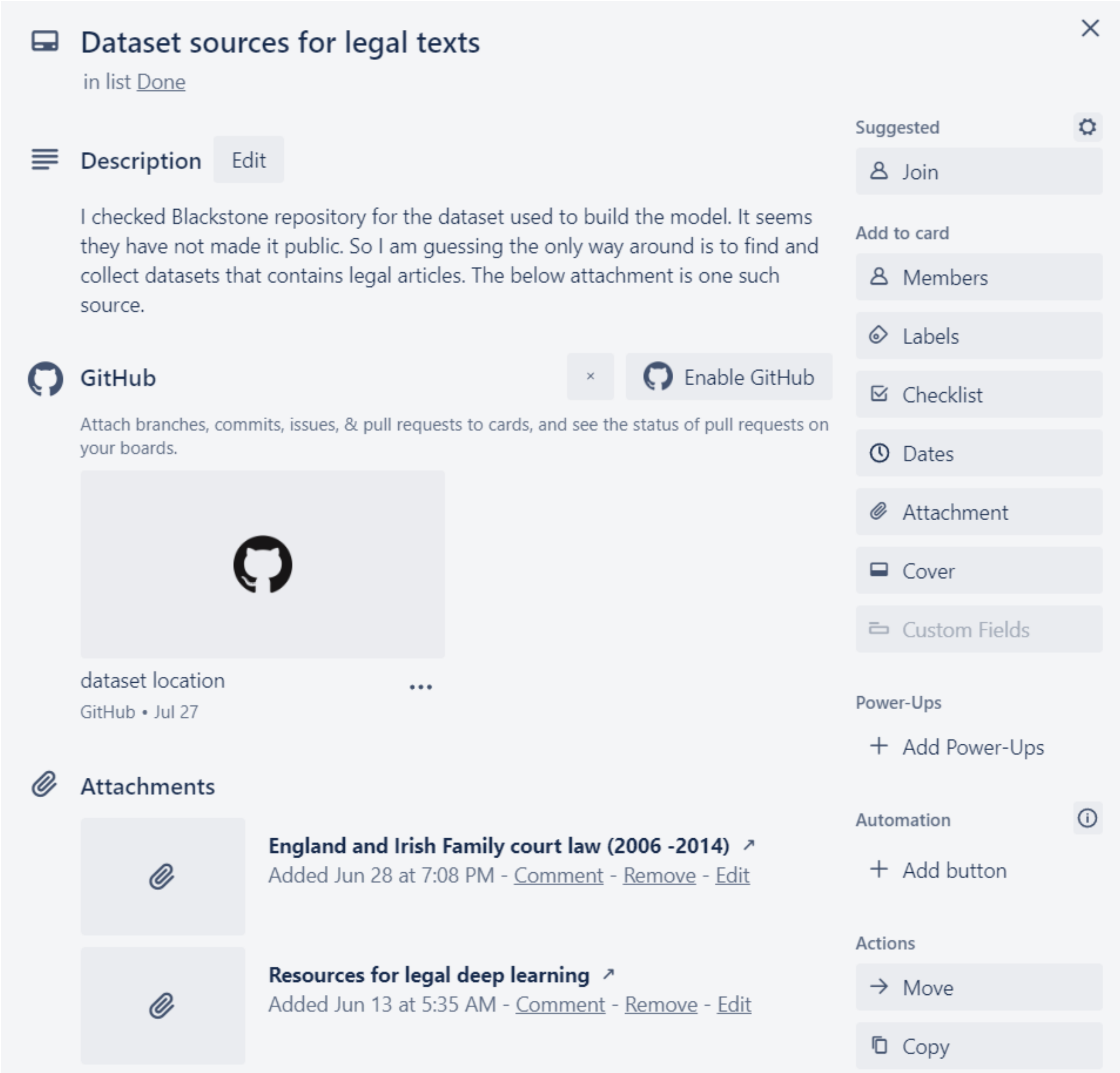

Figure 15-Trello Board task card part 1

Chapter 5: Results

5.1 Framework initial structure

It was concluded that it would be best to understand the neural networks before implementing the models. Based on relevance for this research the three neural network that showed similarity to the hypothesised models where convolutional neural network, Recurrent neural network, and transformers. Therefore, investigated each neural network and deduced their pros and cons. After analysing the pros and cons deduced an initial framework. That the structure should constitute an NER model which incorporates convolutional neural network, question and answer framework that incorporates transformer foundation and a graphical user interface to connect these components into a single system. As a result, the Framework's basic structure was determined.

5.2 Implementing the framework (NER)

Background study on NER model required learning, how to teach a model to identify various entities, how to use the model to forecast these entities from test point of view and how can the end user benefit from this model. The spacy module is the current trend when it comes to creating NER modules. It is known to be efficient and seamless to implement. Spacy is known to be built on convolutional neural network with a few alterations (model?, 2022).

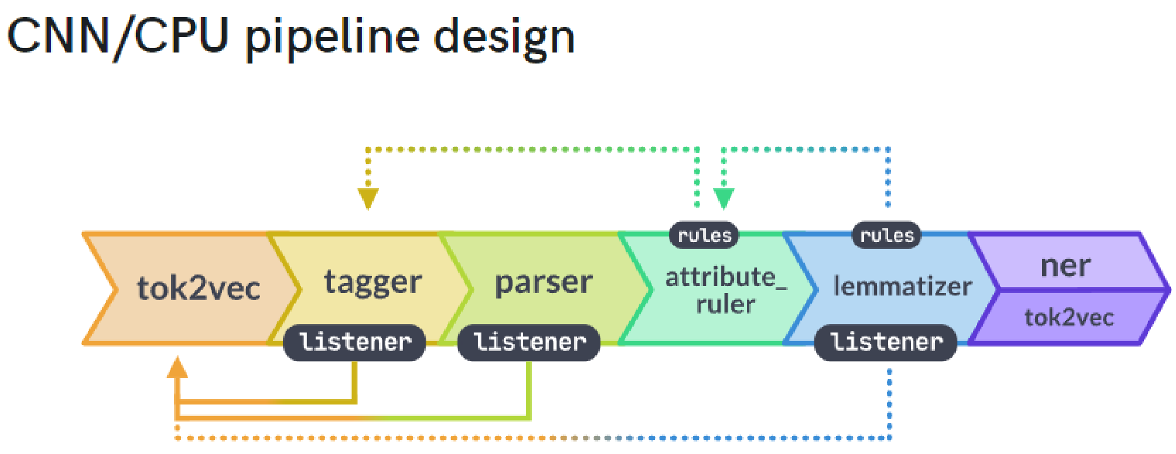

Figure 16-(Trained Models & Pipelines, 2022)

Studies showed that the model is fed with text which contains these entities and that is how the model is taught to recognise entities. Before the dataset can be utilised in the NER model it required to be annotated. Annotation is the process of identifying entities in plain text and labelling them with their respective tags. The dataset that is covered in this study is from a legal background. Hence the tags that needs to be assigned need to be linked with legal terminologies. This was implemented using an annotation tool that is accessible via a website (NER Annotator for SpaCy, 2022). This annotation tool accepts text files and produces a json file containing the entities and the texts from which they were extracted. If the annotation is incorrect it can mess up the accuracy of the NER model that is being build. To understand which entities to pick, went through a plethora of research papers delving into the methodology of annotation for named entity recognition. Those research papers helped in understanding which entities needed to be picked. Took up help from an expert in the field of legal domain when it came to identifying the entities and labelling them as per the legal lingo. The annotation tool partitions the input text based on different special characters. For this research the delimiter that was set for partitioning was the new line. This process was a lot of grunt work, and it took plenty of time before the annotated dataset was generated.

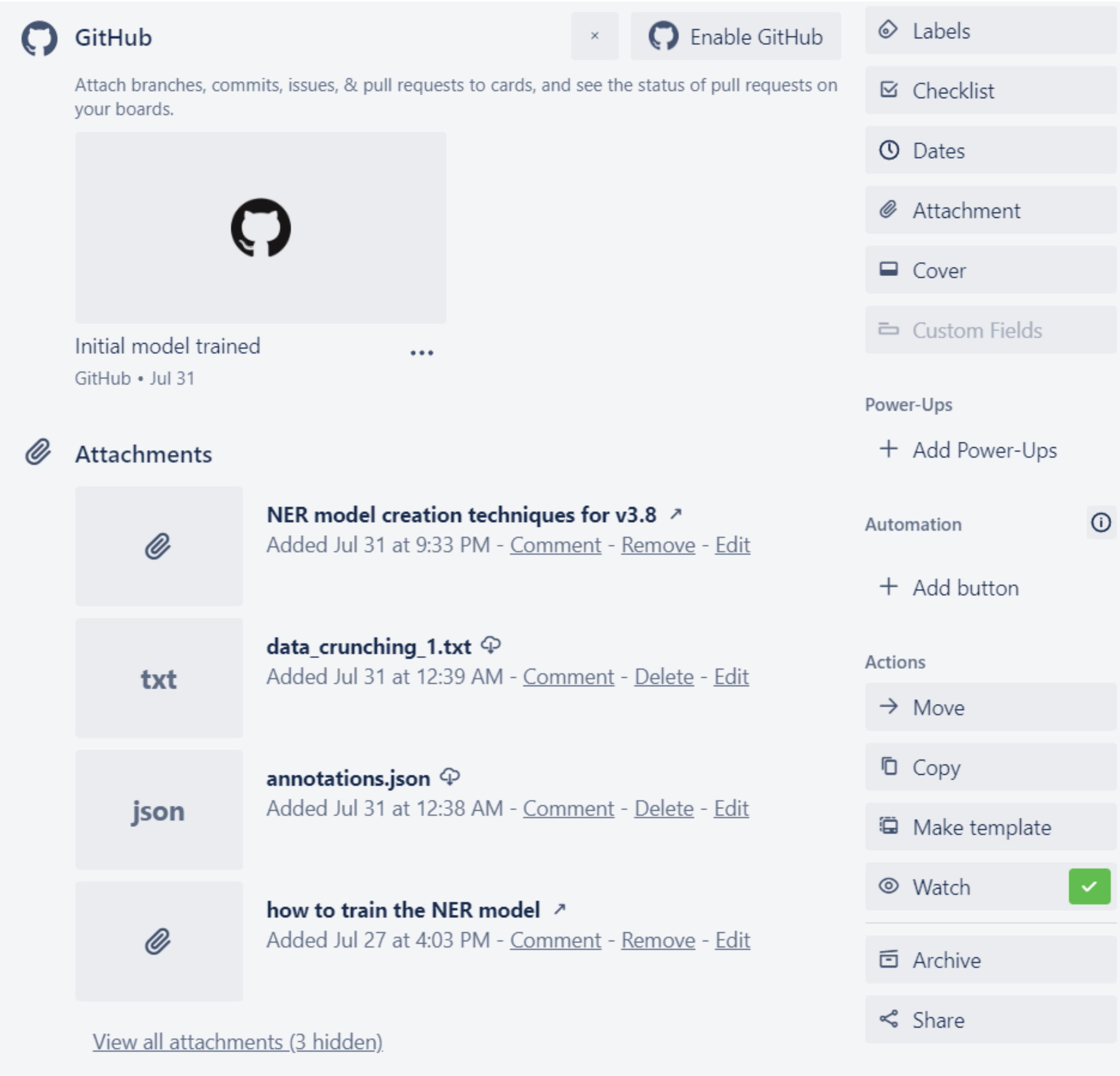

Figure 17-Trello Board task card part 2

5.3 Simulating the framework (NER)

The dataset that was obtained for this research came in chunks. The dataset had to be pre-processed since the blank language class only accepts training data in one go. Before the train dataset is delivered to the blank language class, an iterative loop was built to combine the chunks to a single dataset. This was done using the inbuilt zip function along with initialisation of the data type to dictionary in a single step.

The way to go about it is to define a blank language class and then layer that class with detailed information on as to how to map each label. Meaning feeding the specific entities with their respective labels into the blank language class. The model is then made to memorize these specific entities in an iterative fashion to improve its recognition capability. Once that is done its loss value is calculated and displayed. This helps the user understand how much more training dataset needs to be provided to increase the performance of the model. The model is then saved to disk so that the entire process that was done need not be repeated when running a test case or when showcasing a demo.

5.4 Implementing the framework (Question and Answer model)

For this the inspiration was taken from the hugging face community (Question answering - Hugging Face Course, 2022). They have a model build specifically for a question-and-answer framework. For building this model the dataset needs to be in a certain format. The dataset needs to contain an id, question, context, and answer. To understand the framework of the model the "squad" dataset was used (The Stanford Question Answering Dataset, 2022). The next step involved generating the training and testing dataset for the legal domain. Created two csv files one for training and the other for testing. One of the mandatory fields required is "id" which is supposed to contain MD5 compression of the question to make each datapoint unique. This was created using a web link that generates MD5 hashes when a string is provided in its dialog box.

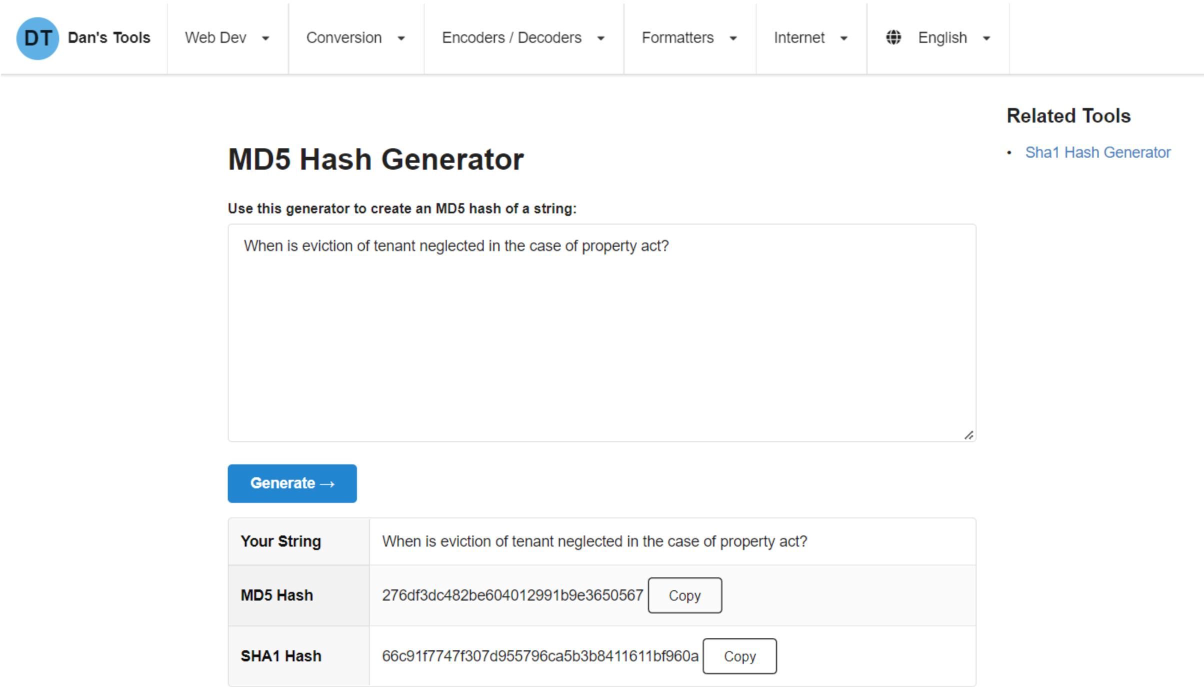

Figure 18-(MD5 Hash Generator, 2022)

The steps in the question-and-answer framework involved tokenizing the dataset along with providing labels for the question and context. The module Auto tokenizer was used along with the pretrained Bert model for effective fast tokenising. It has the feature to store the answer in the context even when truncation is applied. It also has the capability to keep track of the answer even when the offset components are put in place.

An integral portion of pre-processing the training dataset in this framework is to use the label mapping to identify the answer. A simple algorithm with the offset mapping and the sequence id generated by the tokenizer is used to identify the exact location of the answer in the context. This is then verified with the theoretical location of the answer.

This concept is clubbed together into a function for a complete pre-processing of the train dataset. Similar function is utilised in the pre-processing of the validation dataset as well, but it does not involve labelling. Due to offset mapping, there is a chance of miss interpreting question and context in the validation dataset when post-processing this is resolved by setting the offset mapping value of the questions to None (Question answering - Hugging Face Course, 2022).

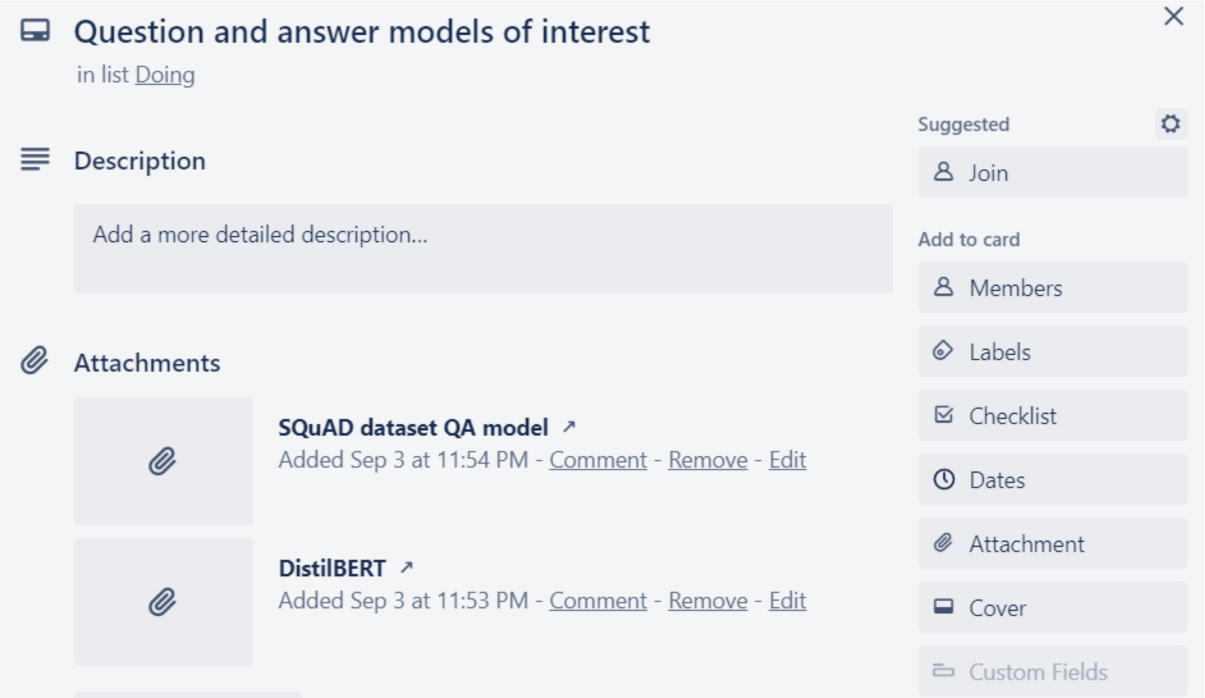

Figure 19-Trello Board task card part 3

5.5 Simulating the framework (Question and Answer model)

In order to help map back the feature to the original context the ID component will be used. This is specifically used to evaluate if the Bert based model is functioning as per expectation. A small set of the validation dataset is utilised to verify if the model is indeed able to predict the answer. Here the ID component is used to map back to the context wherein it helps in identifying the location of the answer.

Chapter 6: Evaluations

For the NER model the validation test dataset is provided directly to the model. Using various attributes already present in the NER, the entities can be displayed from the new set of data. The IDE that is used for this project is PyCharm and it does not have compatibility with the displacy function. When using in jupyter notebook the displacy function can display the entities present in the validation test dataset better.

The result generated by the NER model is showcased below:

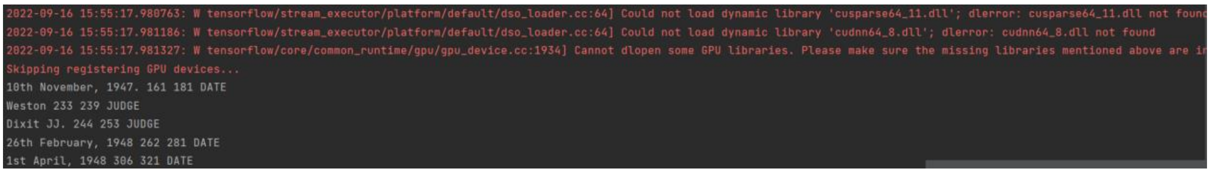

Figure 20-Pycharm IDE terminal NER output

It is accurately able to annotate but the score of its performance is low as the amount of legal dataset that can be manually annotated for this model was limited by human constrains. The limited time frame for the completion of this research and the constraints due to the effort required for building the other models also played a huge factor. The performance of the model taking into account these constrains shows that the framework is in fact solid and can be built upon in future with more viable datasets to strengthen.

In the final steps of the question-and-answer model default tensor flow dataframe is generated using one of the modules called DefaultDataCollator (Question answering - Hugging Face Course, 2022). The pre-processed training and validation dataset are modified using the command to_tf_dataset which takes in the default tensor flow dataframe to create a tensor flow dataframe based on the contents on the train and validation dataset. Followed by the creation of the optimizer using set values for learning rate and weight decay rate. Then the model is compiled and fitted, which is then saved to disk via using one of the inbuilt features. This model is then used for generating predictions. To exhibit the model's capability, it was vital to save the model instead of training and fitting the model in a repeated loop. The inbuilt attribute of the tensor flow question and answer model was used to save it and later load it in the case the model was already created. This small sequence of steps helped break the redundant cycle of generating, training, and fitting of the model.

Finally, after the model is generated and fitted. It is used to generate the result for the test dataset. This portion of the script is the backbone of this research. The module transformers created by the hugging face community has attribute called pipeline. Which is utilised to load a check point of the model to provide a framework to enter input of interest and generate the output without having the constraint of the format for the data.

The result generated by the question-and-answer model is showcased below:

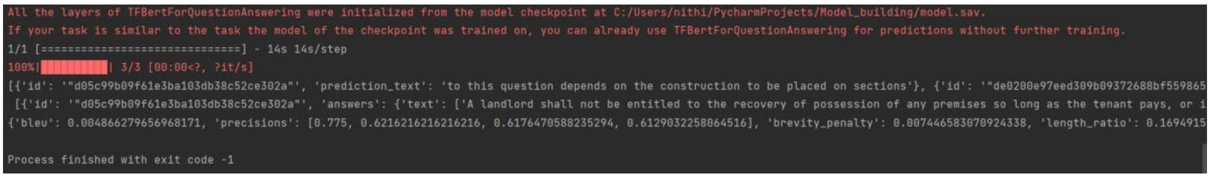

Figure 21-Pycharm IDE terminal Question and Answer model output

The compute metrics can be utilised for visually observing the model's performance. Initially the metrics using the squad framework was utilised but since it did not give a clear understanding of the match percentage the BLEU metric was utilised. This method checks the similarity between the predicted text and the reference text. BLEU (Bilingual Evaluation Understudy) measures the accuracy of material that has been machine converted from a basic language to another (BLEU - a Hugging Face Space by evaluate-metric, 2022). This is used at the end of the question-and-answer model to evaluate the model prediction score by using the reference dataset against the predicted values. Quality here is defined as the similarity between machine and human's text generation capability. One of the first measures to assert a strong association with human judgments of quality was BLEU, which is now one of the most widely used and affordable metrics (BLEU - a Hugging Face Space by evaluate-metric, 2022).

This model also required the legal dataset to be pre-processed and in a specific format for the model to be build. Grunt work was done to create as much datapoints as possible in the specific format of question, context, and answer. The human constraints of manually generating datapoints is reflected in the performance of the model.

The User interface that was developed was deployed locally in a virtual server and evaluated on the basis of its functionality. The below figure gives an overview of the UI framework. For this instance it is able to correctly identify the answer to the query on the basis of the context.

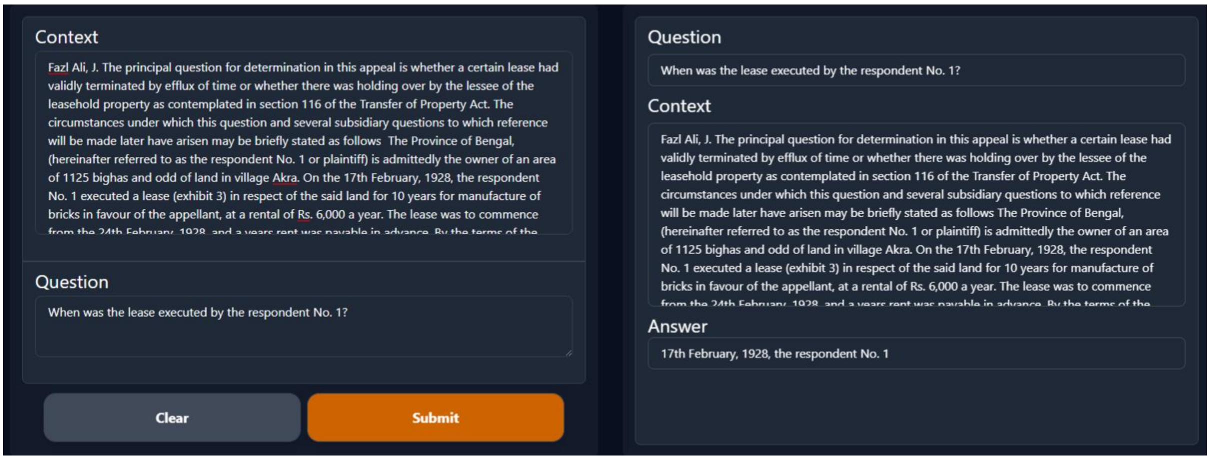

Figure 22-Flask web app prototype

Chapter 7: Reflections

7.1 Limitations

The issues that were faced while managing this research are as follows

7.1.1 Dataset issues

Due to the tedious labour needed to generate the dataset for the NER and the question-and-answer model. Less quantity of dataset was producible, leading to high loss value and bad performance from the models.

7.1.2 GPU issue

The question-and-answer model required GPU. For python to recognise the GPU. CUDA and CuDNN latest versions were installed. This did not resolve the issue of GPU as incompatibility of the latest version of CUDA with the GPU resulted in memory fragmentation error. The system was not able to allocate memory properly when the model fitting was run.

7.1.3 GPU Time limit issue

During the development stage there was the requirement of hosting the virtual server for longer periods which lead to google colab restricting the usage of GPU. This meant the model had to be run on the google colab CPU leading to crash when fitting the model.

7.1.4 Local port issue

Since the IDE PyCharm was scrapped and colab was made as the primary resource for the execution of the tool. When building flask for the web app, ran into the issue of not able to host a local server.

7.1.5 CORS error

Once the work around for hosting in local port was resolved. CORS error was observed as the hosting was done via a virtual server. When trying to obtain response from the virtual server the chrome browser generated CORS error. This occurs when the browser security denies response send to the virtual server as it recognises it as malware.

7.2 Solutions

Few of the issues were resolved by dedicated perseverance and hard work while some remained a persistent issue.

7.2.1 GPU issue debugged

To resolve this issue switched from local system's IDE to google colab which provided the feature of GPU runtime with specific limit.

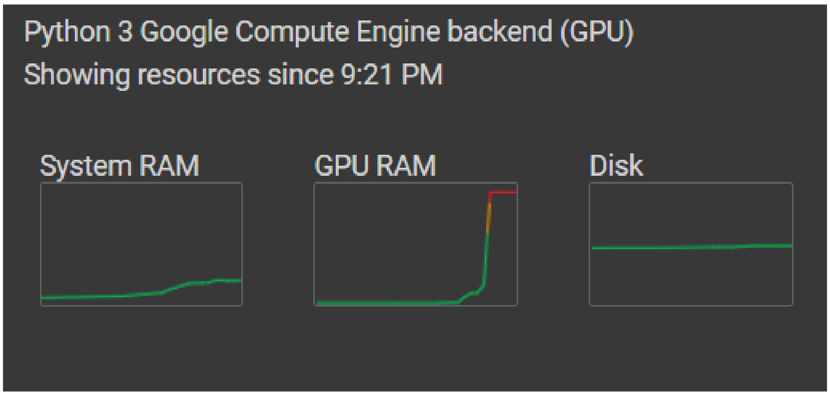

Figure 23-Google colab resources view

7.2.2 Local port issue debugged

To resolve this issue utilised the module called flask_ngrok which helped to run a virtual server via google colab.

```
# from flask_restful import Api, Resource

app = Flask(__name__)
# CORS(app, support_credentials=True)
# api = Api(app)
run_with_ngrok(app)


@app.route('/predict',methods=['POST'])
```

Figure 24-Google colab flask_ngrok syntax

```
 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Running on http://133b-35-197-149-196.ngrok.io
 * Traffic stats available on http://127.0.0.1:4040
```

Figure 25-Google colab terminal virtual server location

7.2.3 CORS error debugged

To resolve this issue utilised a *.bat file with specifications to disable chrome browser security. The *.bat file contains the following content:

```
"C:\Program Files\Google\Chrome\Application\chrome.exe" --disable-web-security --user-data-dir=~/chromeTempRepo
```

Figure 26-.bat file content*

7.3 Future works

The concept of utilising partially annotated dataset for training and building the NER model is something that can be looked into as a future scope. The build constructed in this study is specifically designed for the English language and an added feature would be to include more languages. The graphical user interface is an area that was not explored much in detail. The modules like PyQt5, PySimpleGUI, Kivy and many others have features that might bring a better visual perspective than the web app that is currently developed. It is an area of study that can be investigated for future works. Another research would be to utilise recurrent neural network to make partial legal dataset wholesome. This could cut down on the amount of time needed to find datasets while creating NLP models. The performance of the model in this study is not up to par due to the limitation of datasets. For the model to produce better results, the insufficient dataset may be improved upon, and this is something to look forward to.

Chapter 8: Business benefits

8.1 Market spectrum

In the current era NER models have well defined entities in place particularly for the English language. This means that the roots of the advancement have spread well and wide in the legal domain as well for such kind of systems. Due to improvements in legal systems, the current achievable usage is as follows (Badji, I., 2018):

- Comparing instances to find similarities for defence
- “Search engines”
- “Anonymizing documents”
- “Automatic generation of contracts”
- “Summarization of documents”
- “Raking lawyers based on the number and types of cases solved” (Badji, I., 2018)

In order to accommodate a larger variety of demands and possibilities, this system is made available while taking into consideration the kind of text, legal papers, and non-formal documents. In this context, a non-formal document is any text file that is not particularly a legal document and could range from a tweet to a remark on a social networking site (Badji, I., 2018).

Based on the research so far, the commercial spectrum of these type of products is not vibrant due to its limitations. The limitations that restrict these tools spectrum are as follows:

- No proper legal entity definition
- The training materials have an impact on the tool's performance.
- A dearth of corpora containing documents with annotations identifying legal entities

Presently on the premises of Ansoff matrix the market spectrum of these gadgets lies in the diversification region. The Ansoff Matrix shows the possibilities available to organisations looking to increase revenue or profitability on a two-by-two grid. It is helpful because it offers a straightforward framework that combines all the possible strategic trajectories for a company into a single analytical instrument. The matrix's axes are focused on the crucial connection that an organisation must manage: the delivery of goods to consumers. Thus, Ansoff matrix depicts the four possible courses of action (Meldrum, M. and McDonald, M., 1995):

- Focusing on current products for current markets.
- Attempting to find new products for current markets.
- Attempting to sell old items in new markets.
- Pursuing new product and market diversification. (Meldrum, M. and McDonald, M., 1995)

Figure 27-(Peterdy, 2022)

The tool that is being build is fairly new and the market for it is not well established. The reason for this being, the market available currently is for tools such as search engines, document summarization, contract generation and lawyer ranking (Badji, I., 2018). But the field that is of interest is a market to educate individuals in legal domain specific knowledge. Hence the product that is being build will be under the diversification grid and the risk level for such a transition is high. As the risk factor is high the reward will be even sweeter as the feet to pull off such unchartered waters is what makes this research stand out. So, to get a bit more insight on the market, the industry's attractiveness needs to be determined. That is where Porter's five forces come into play as it is based on the five competitive factors:

- Threat of entry
- Threat of substitutes
- Bargaining power of the buyers
- Bargaining power of suppliers
- Extent of rivalry between competitors (Whittington et al., 2020)

The threat of entry:

When entrance barriers are high, the threat of entry is low, and the opposite is also true (Whittington et al., 2020). The usual barriers are as follows:

- Economies of scale or high fixed cost
- Experience and learning
- Access to supply and distribution channels
- Differentiation and market penetration costs
- Legislation or government restrictions (Whittington et al., 2020)

For this research the barriers are low hence the threat of entry is high. The barriers in this field would be experience and learning, access to supply and distribution channels, and differentiation and market penetration costs (Whittington et al., 2020). Since most of the material that can help in development of such applications is open source, experience and learning can be attained without much difficulty. Access to supply is also achievable as there are prebuilt models which can be obtained easily with minimal search in the net. Market penetration cost is not present as there is no strong organisation that have currently invested in this market sector. As a result, threat of entry is at a colossal high.

Threat of substitutes:

Products and services that give comparable advantages to those provided by others in the sector through a different technique are known as substitutes (Whittington et al., 2020). The reason for customers choosing an alternative is due to the following:

- Price or performance ratio.
- An invention that raises consumer happiness helps the substitute.
- The extra industry effects: In other words, if switching costs are low, the danger level will rise, and the industry's allure will decline (Whittington et al., 2020)

The performance ratio is something that cannot be determined at present as the domain of this research is still in the developing phase. There can be new techniques that will be introduced over time, and it can be a substitute based on its performance. Threat of substitute is also high.

Bargaining power of buyer’s:

Buyers are not always the final consumers of an organisation, but they are its immediate customers. Powerful consumers may demand lower pricing or better product services, which would reduce profitability (Whittington et al., 2020).

The initial concept of marketing this product is by introducing it as a broad teaching aid for the legally uneducated individuals. So, the direct consumers will be the buyers. Once the product is launched, it will need ongoing integration as and when customers deem it necessary, which in reality increases the buyer's negotiating power.

Bargaining power of the suppliers:

Organizations that need to generate goods and services are known as suppliers. Strong suppliers might reduce a company's earnings (Whittington et al., 2020). Since the majority of the essential components for the product under discussion are constructed utilising open sources, there are no suppliers.

Competitive rivalry:

Competitive rivals are businesses that offer comparable goods and services to the same clientele and compete directly with one another in the same sector (Whittington et al., 2020). Rivalry becomes more intense when:

- Competitors are of similar size and capability
- Rivals are aggressive when seeking out leadership
- When the market keeps maturing or declining
- High fixed cost
- Exit barriers are high
- Low level of differentiation (Whittington et al., 2020)

Other factors that affect rivalry are:

- Shake out as industry growth slows during the early stages of development
- limited growth because of a saturated market

- decline when innovation and shifting environmental conditions decrease demand for industrial products and services (Whittington et al., 2020)

Since there are now no commercially available tools that can accomplish the exact tasks stated in this study and the product development in this research is still in its early stages. It is safe to say that the product has no competitors at this time.

8.2 Related works that are commercialised

This study can be envisioned into a product that could be sold as a broad teaching aid for the illiterate in this domain. Also, as an application to large organisations dealing with legal matters as the first point of contact to address routine inquiries. This can cut down on expenditure and valuable time that is lost when consulting law firms. Which raises the question of related works that can be found in this domain. The recent product that was commercialised in this domain is by the organisation called Lynx. It's a small medium sized enterprise that provides services for accessing legislation, case law, standards, industry norms, and best practises among the vast volumes of digital regulatory compliance materials. The main objective of the organisation is to deal with legal and regulatory compliance data. In Layman's terms Companies that want to enter a new market must follow sector-specific procedures, various standards, and applicable laws. Lynx offers services that can streamline these laborious tasks, which typically call for field expertise. It's basically a cloud service eco system which provides services for the aforementioned applications. Even though it has a plethora of services it does not provide features that is similar to this research study (Lynx - summary, 2022).

Hence this does justify why this research has a competitive advantage. Competitive advantage refers to the factor that provides an edge to the product or the organisation against its rivals.

Chapter 9: Conclusion

This study has helped in understanding the difficulty in procuring legal dataset. Got Educated on the various instruments available at the disposal for annotating dataset in the format a natural language programming model accepts. Experienced the restrictions on model construction caused by the need for datasets in a certain format. Due to the grunt work required for such specs identified different alternatives to bypass these steps. One such alternative was using partially annotated datasets to educate the model. The complexity of this procedure and time constraints gave the final blow to drop this approach. Learned different theories behind the neural networks that were being constructed. For instance, during research, it was discovered that the convolutional neural network combined with the transformers produces the spacy module utilised to develop the NER model. Furthermore, the practicality of the theoretical concepts relating to the neural network were put to test when building every model. Some notions of the theoretical knowledge were exceptionally helpful when constructing these models while some still remained a mystery.

The first research publication on questions and answers made the observation that the NER model would come in handy when building the question-and-answer (QA) framework. It elaborated on how the entities can be utilised to identify the answer to the query. A part of this concept is still in play when it comes to the model that was constructed. On how to build a QA model came from the hugging face community. The researchers in the community provided an amazing guide on how the model can be built and what are the parameters used to build such an intricate model. This provided the insight required to mould the dataset according to the requirements of the model. Identified the different functions that was built based on simple algorithms that take up small task to transform input into well-defined results. Even when the complexity of the code increased the roots of the algorithm that gave birth to this complexity helped in moving forward with each step.

To evaluate each model the loss was calculated. Even though the value obtained for the loss was nowhere near the hypothesised expectation. The joy in building, learning and finally bring the models to fruition helped forget the limitations that came with the setup. A lot of manual work went into the creation of the input dataset, but it still yielded less quantity when compared to the size that a model expects for the system to grasp the predictive algorithm. Hence the metric measures aren't that far off. To get an understanding of the measurement that can provide information on the model's performance the bleu evaluation algorithm was deployed. Which yielded good results when compared one to one against its counterparts but not so well when compared against the whole. Development of the web application went side by side with the development of the model. Obtained knowledge on the modules like flask, flask_ngrok, flask_restful and some basic html programming when it came to developing the web application. Grasped the concept to integrate the models with frontend using the virtual server deployed using the flask module. Developing the debugging skills that was dormant has been considered the key take away from this project.

Chapter 10: Supervision diary

Schedule	Tasks
12th July	**Data Pre-processing**
19th July	**NER model construction**
26th July	**Annotation tool**
2nd August	**Table of contents**
9th August	**Literature review initial draft**
16th August	**Current progress**
6th September	**Question and Answer model**
13th September	**Debugged QA model issues**
20th September	**Final draft**

Credits

Indian legal dataset:

Malik, V., Sanjay, R., Nigam, S.K., Ghosh, K., Guha, S.K., Bhattacharya, A. and Modi, A., 2021. ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. *arXiv preprint arXiv:2105.13562*.

Question and Answer model:

Huggingface.co. 2022. *Question answering - Hugging Face Course.* [online] Available at: <https://huggingface.co/course/chapter7/7?fw=tf> [Accessed 17 September 2022].

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M. and Kim, B., 2018. Sanity checks for saliency maps. *Advances in neural information processing systems*, *31*.
2. Almeida, F. and Xexéo, G., 2019. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*.
3. Badji, I., 2018. *Legal entity extraction with NER systems* (Doctoral dissertation, ETSI_Informatica).
4. Bahdanau, D., Cho, K. and Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
5. Bailii.org. 2022. *BAILII - Reproduction and Copyright*. [online] Available at: <https://www.bailii.org/bailii/copyright.html> [Accessed 23 September 2022].
6. Bengio, Y., Simard, P. and Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), pp.157-166.
7. Bernhardt, G., 2022. *Build an N-Gram Text Analyzer for SEO using Python | Advanced Python SEO Scripts for Next Level Results- importSEM.* [online] Advanced Python SEO Scripts for Next Level Results- importSEM. Available at: <https://importsem.com/build-an-n-gram-text-analyzer-for-seo-using-python/> [Accessed 20 September 2022].

8. Blog.fastforwardlabs.com. 2022. *How to Explain HuggingFace BERT for Question Answering NLP Models with TF 2.0.* [online] Available at: <https://blog.fastforwardlabs.com/2020/06/22/how-to-explain-huggingface-bert-for-question-answering-nlp-models-with-tf-2.0.html> [Accessed 17 September 2022].
9. Braşoveanu, A.M. and Andonie, R., 2020, September. Visualizing transformers for nlp: a brief survey. In *2020 24th International Conference Information Visualisation (IV)* (pp. 270-279). IEEE.
10. Brownlee, J., 2022. *Best Practices for Text Classification with Deep Learning.* [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/best-practices-document-classification-deep-learning/> [Accessed 20 September 2022].
11. Brunner, G., Liu, Y., Pascual, D., Richter, O., Ciaramita, M. and Wattenhofer, R., 2019. On identifiability in transformers. *arXiv preprint arXiv:1908.04211*.
12. Chang, M.W., Ratinov, L. and Roth, D., 2007, June. Guiding semi-supervision with constraint-driven learning. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 280-287).
13. Clark, K., Khandelwal, U., Levy, O. and Manning, C.D., 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.
14. Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
15. Gonen, H. and Goldberg, Y., 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *arXiv preprint arXiv:1903.03862*.
16. Hochreiter, S., 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), pp.107-116.
17. Huggingface.co. 2022. *BLEU - a Hugging Face Space by evaluate-metric.* [online] Available at: <https://huggingface.co/spaces/evaluate-metric/bleu> [Accessed 17 September 2022].
18. Huggingface.co. 2022. *Question answering - Hugging Face Course.* [online] Available at: <https://huggingface.co/course/chapter7/7?fw=tf> [Accessed 17 September 2022].
19. ICLR&D. 2022. *Case Genie goes live!.* [online] Available at: <https://research.iclr.co.uk/blog/blackst%20one-goes-live> [Accessed 17 September 2022].
20. Jain, S. and Wallace, B.C., 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186*.
21. Järvelin, K. and Kekäläinen, J., 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, *20*(4), pp.422-446.
22. Jennifer Kwentoh. 2022. *Part-of-Speech Tagging examples in Python - Jennifer Kwentoh.* [online] Available at: <https://jenniferkwentoh.com/part-of-speech-tagging-examples-in-python/> [Accessed 20 September 2022].
23. Johnson, G., Whittington, R., Regnér, P., Angwin, D. and Scholes, K., 2020. *Exploring strategy*. Pearson UK.
24. Kien, P.M., Nguyen, H.T., Bach, N.X., Tran, V., Le Nguyen, M. and Phuong, T.M., 2020, December. Answering legal questions by learning neural attentive text representation. In *Proceedings of the 28th International Conference on Computational Linguistics* (pp. 988-998).
25. Langford, I., 2009. Fair trial: The history of an idea. *Journal of human rights*, *8*(1), pp.37-52.
26. Lynx-project.eu. 2022. *Lynx - summary.* [online] Available at: <https://lynx-project.eu/project/summary> [Accessed 17 September 2022].
27. Malik, V., Sanjay, R., Nigam, S.K., Ghosh, K., Guha, S.K., Bhattacharya, A. and Modi, A., 2021. ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. *arXiv preprint arXiv:2105.13562*.
28. Mayhew, S., Chaturvedi, S., Tsai, C.T. and Roth, D., 2019. Named entity recognition with partially annotated training data. *arXiv preprint arXiv:1909.09270*.
29. McCann, B., Bradbury, J., Xiong, C. and Socher, R., 2017. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, *30*.

30. Md5hashgenerator.com. 2022. *MD5 Hash Generator.* [online] Available at: <https://www.md5hashgenerator.com/> [Accessed 17 September 2022].
31. Medium. 2022. *Part Of Speech Tagging for Beginners.* [online] Available at: <https://towardsdatascience.com/part-of-speech-tagging-for-beginners-3a0754b2ebba> [Accessed 20 September 2022].
32. Meldrum, M. and McDonald, M., 1995. The Ansoff Matrix. In *Key marketing concepts* (pp. 121-126). Palgrave, London.
33. Merritt, R., 2022. *What Is a Transformer Model?.* [online] NVIDIA Blog. Available at: <https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/> [Accessed 17 September 2022].
34. model?, W., 2022. *Which Deep Learning Algorithm does Spacy uses when we train Custom model?.* [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/60381170/which-deep-learning-algorithm-does-spacy-uses-when-we-train-custom-model> [Accessed 17 September 2022].
35. Nay, J., 2018. Natural language processing and machine learning for law and policy texts. *Available at SSRN 3438276*.
36. NewsCatcher. 2022. *How To Train Custom Named Entity Recognition[NER] Model With SpaCy - NewsCatcher.* [online] Available at: <https://newscatcherapi.com/blog/train-custom-named-entity-recognition-ner-model-with-spacy-v3> [Accessed 17 September 2022].
37. Paruchuri, V., 2022. *Tutorial: Web Scraping with Python Using Beautiful Soup.* [online] DATAQUEST. Available at: <https://www.dataquest.io/blog/web-scraping-python-using-beautiful-soup/> [Accessed 17 September 2022].
38. Pascanu, R., Mikolov, T. and Bengio, Y., 2013, May. On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310-1318). PMLR.
39. Peterdy, K., 2022. *Ansoff Matrix.* [online] Corporate Finance Institute. Available at: <https://corporatefinanceinstitute.com/resources/knowledge/strategy/ansoff-matrix/> [Accessed 20 September 2022].
40. Pythonbasics.org. 2022. *What is Flask Python - Python Tutorial.* [online] Available at: <https://pythonbasics.org/what-is-flask-python/#:~:text=Flask%20is%20a%20web%20framework,like%20url%20routing%2C%20template%20engine.> [Accessed 20 September 2022].
41. Rajpurkar.github.io. 2022. *The Stanford Question Answering Dataset.* [online] Available at: <https://rajpurkar.github.io/SQuAD-explorer/> [Accessed 17 September 2022].
42. Resnik, J., 2011. Bring Back Bentham:" Open Courts,"" Terror Trials," and Public Sphere (s). *Law & Ethics of Human Rights*, *5*(1), pp.4-69.
43. Salton, G., Wong, A. and Yang, C.S., 1975. A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), pp.613-620.
44. Spacy.io. 2022. *Trained Models & Pipelines.* [online] Available at: <https://spacy.io/models#conventions> [Accessed 21 September 2022].
45. Tecoholic.github.io. 2022. *NER Annotator for SpaCy.* [online] Available at: <https://tecoholic.github.io/ner-annotator/> [Accessed 17 September 2022].
46. Turian, J., Ratinov, L. and Bengio, Y., 2010, July. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 384-394).
47. Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A.N., Gouws, S., Jones, L., Kaiser, Ł., Kalchbrenner, N., Parmar, N. and Sepassi, R., 2018. Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*.
48. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, *30*.
49. VEEN, F., 2022. *The Neural Network Zoo - The Asimov Institute.* [online] The Asimov Institute. Available at: <https://www.asimovinstitute.org/neural-network-zoo/> [Accessed 20 September 2022].

50. Veluri, Z.A.R., Named Entity Recognition and Question Answering Using Word Vectors and Clustering.
51. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. and Davison, J., 2020, October. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (pp. 38-45).
52. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. and Davison, J., 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
53. Yilmaz, Z.A., Wang, S., Yang, W., Zhang, H. and Lin, J., 2019, November. Applying BERT to document retrieval with birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations* (pp. 19-24).

Bibliography

1. Joseph,N.M., 2022, Interpreting legal texts using natural language processing

Appendices

The modules that need to be installed are explicitly provided in the Requirements.txt file. Chrome, Visual studio, Angular cli, Thunder client and node.js needs to be installed for the web application to work. The NLP_NER_QA_FE_BE.ipynb file contains the project main code. The input to the main code is annotated_datasets folder, train.csv and test.csv. The .ipynb file can be loaded up in google colab. The GPU resource needs to be switched on before execution of the code. The visual studio platform needs to be opened with NLP_frontend folder which contains the frontend script. Once the main code is run, it will generate a virtual server IP that needs to be copied to the services file in visual code. Furthermore, command line needs to be run with the command `ng serve` to execute the development server. To load up the development server IP in chrome browser the chrome security needs to be disabled. *.bat file containing chrome disable security command needs to be executed with administrator privileges. After that the development server IP provided in the command line needs to be entered on the search bar to open up the user interface for the legal query engine. Provide the query and context in the dialog box pertaining to them. Then pressing submit will provide the models answer.