Legal Question Answering System

Third Year Project

Karan Kapotra

BSc (Hons) in Computer Science and Mathematics

Supervised by Dr. Bijan Parsia

Acknowledgements

Firstly, I would like to thank my supervisor Dr Bijan Parsia for introducing me and my project teammates to the legal technological landscape. Each encounter with him was full of interest and learning. His lessons were not limited to academics either and I will remember him as one of the best supervisors I had the pleasure of being mentored by.

Further gratitude goes towards my teammates, Ana-Silvia Serban and Khesim Reid. Both of them from the very start had an extreme desire to learn and succeed, which I still look up to. They took each challenge head-on and never let me feel like an obstacle was too much. They have supported my learning and understanding from the very beginning and I am happy that these two brilliant computer scientists are the ones I can call my teammates and, more importantly, my friends.

Contents

1 Abstract

This report discusses the process and research that went into creating a system which allows a user to ask a question over a corpus of documents. An argument for this system will be presented first, giving reasons as to why this system is needed and effective. I will discuss the design and implementation, evaluating its benefits and drawbacks in relation to the objectives of the system. I will strive to compare it with other similar systems, which aim to solve similar problems.

2 Background

This section briefly describes the origin of the motivation for this project and the different ways technology has been used in the legal industry so far. This is coupled with some preliminary research I completed around the subject to understand the scale and what kind of impact technology and software changes can have on this important industry.

Before I started this project, I had the chance to attend 'The Shifting Regulatory and Legal Technology Landscape' event at Bloomberg London. There I met the Regional Head of EMEA Legal Negotiations Darya Solovey. A conversation with her led to the conversation of the impact of technology in her profession. This is where my interest in the legal technology landscape began and made me think about the uses of technology in an industry which has otherwise been slow to adopt innovative means of operations [1]. We spoke about the different ways she has implemented technology in her department, including software which finds loopholes in law contracts and other legal correspondence. As a result of this meeting and event, I began looking into other ways technology can and has been used in the legal industry as a starting point for my project.

Technology in the legal industry has seen a boom in recent years with many firms looking to find more efficient ways to improve their processes and lengthy operations. Legal tasks comprise collecting and working with large amounts of data of different forms. Technology in different forms leads to the formation of innovative solutions for these arduous tasks.

For instance, AI and machine learning have been used for an array of tasks, such as predictive coding. This is a form of technology-assisted review used to assess the relevance of vast numbers of documents for purposes of electronic disclosure (e-disclosure), which I will expand on further. Predictive coding - mandated in certain cases following Brown v BCA Trading and others [2016] EWHC 1464 (Ch) - employs a combination of keyword search and iterative computer learning to rank the relevance of each particular document [2].

Another use of technology in the legal industry is cloud computing. Most industries have seen a shift towards implementing software engineering techniques into their day-to-day working habits, supported with individual and team-based software, such as Microsoft Teams and Slack. Software like the aforementioned and the abundance of internet connectivity allows for many people to work remotely, attracting young talent who are more inclined to this type of work. Aside from this, many firms are also realising that moving their IT infrastructure to cloud services allows for improved cost efficiencies [2].

Automation is a simple but effective use of technology in the law firms which have sought to improve their efficiency. Tasks such as billing, and other such administrator affairs are routine and time-consuming, therefore the use of automation software negates this and allows for lawyers to better use their time with clients [2].

Chatbots are one of the most user-friendly technologies of recent times. It gives the user the experience of speaking to a person with the exact knowledge the client might be seeking. A chatbot can range from accepting to binary answers to extrapolating answers from colloquial statements from a range of languages. This is particularly applicable for the legal industry since there are a lot of interactions with clients who may not be technology averse, therefore using such user-friendly software can increase client satisfaction. A user interface based on a chatbot is present in my project as a means to make it more understandable for the intended users, which in our case are the lawyers [2].

Researching these different uses of technology allowed me to gain an understanding of which requirements the legal industry and law firms had when it came to technology and what mindset was needed when approaching this project.

3 Introduction

Law firms have mostly operated in a standard way for generations. It is an incredibly old profession with deep-rooted sentiments in paperwork and long hours. Nevertheless, in a world where technology is rapidly challenging each industry, law firms have noticed and have slowly started to invest their resources also. Change in client requirements, demand in better and efficient value pushes law firms to invest in more and more technology to keep up with the competition. For law firms to invest in technology, they have to find the areas in which this investment will have the most impact and profit. For this reason, I have chosen to focus on two specific areas of a law firm's usual operations: e-disclosure and e-discovery. For many lawyers, one of the most time consuming and inefficient tasks are associated with data collecting, processing and delivering. The data they work with is usually in many different forms, which makes the task even more difficult and lethargic. We want to understand what these tasks are in detail before we can start to create the solutions and the system for it. We can break down this task into two specific terms: e-disclosure and e-discovery. These terms are not limited to the legal industry, however, both of them are highly prevalent in legal proceedings due to the fact that the job demands that such due diligence is carried out.

First, we will describe what each of these terms means, their role in procedures and what exactly makes them such a difficult process. We will evaluate the use of automation for each of them and relate them to the decisions made in my project. We will be describing them mostly in a legal setting as this is what our project pertains to.

3.1 E-Discovery

E-discovery refers to a process in which legal personnel, such as lawyers or paralegals carry out "discovery" of electronically stored or transferred data, related to a particular case or investigation. "Discovery" is a legal term defined as "...in which each party, through the law of civil procedure, can obtain evidence from the other party or parties by means of

discovery devices such as interrogatories, requests for production of documents, requests for admissions and depositions." [4]. This still holds for the definition of e-discovery, however only encompasses electronic data as previously stated. It is clear to see why this task is so important and why the software, which may be created to automate or assist this task has to be reliable and tested. In a legal case, the data has to be pertinent and complete to answer the investigations and questions that have or may arise [3]. The lack of information can lead to large losses in profits or even the difference between life and death. The system design, therefore, has to not allow for any mistakes. As it is also a legal proceeding, the design must be transparent. This means that the flow of data must be traceable and therefore accountable and non-refutable. This allows the data to be verified and usable. However, the data must also be secure and safe as it most likely going to be confidential. Therefore, the system must be able to proven secure and tested for this. However, the question then arises, what if the system makes a mistake? The system cannot be "blamed" therefore the users and creators will have to suffer any consequences in a situation where the system is the fault, such as producing the wrong output or missing critical information. These requirements and thoughts are all that I took into account while designing the system.

3.2 E-Disclosure

To describe e-disclosure I will be pertaining to the official UK government guidelines and extrapolating the guidelines for this task from this source. The definition of legal disclosure is described by the legal firm Pinsent Masons as "...stage of the litigation process when each party is required to disclose the documents that are relevant to the issues in dispute to the other party." Similar to e-discovery, the definition doesn't change, however only encompasses electronic data when referred to as e-disclosure. The UK government defines the following as the guidelines under which e-disclosure should be carried out [5]. "When considering disclosure of Electronic Documents, the parties and their legal representatives should bear in mind the following general principles –

(1) Electronic Documents should be managed efficiently in order to minimise the cost incurred;
(2) technology should be used in order to ensure that document management activities are undertaken efficiently and effectively;
(3) disclosure should be given in a manner which gives effect to the overriding objective;
(4) Electronic Documents should generally be made available for inspection in a form which allows the party receiving the documents the same ability to access, search, review and display the documents as the party giving disclosure;
(5) disclosure of Electronic Documents which are of no relevance to the proceedings may place an excessive burden in time and cost on the party to whom disclosure is given."

3.3 Requirements for the System

From these guidelines, we can extrapolate which requirements our system will need to satisfy along with a better understanding of how it should act. The first guideline states that the documents should be managed in such a way such that cost is kept minimal. This means that our system should, again, be transparent and allow the user to handle the data efficiently, thus not using unnecessary time and money. The second point is the most interesting because it shows that the government guidelines recognise that technology is proven to be useful and they make the process more efficient and effective compared to traditional methods. The fourth point states that the data and documents transmitted to the different parties should be available for inspection in the same manner as the original party. This means our system has to not be biased towards any party and give the same results for the same inputs. The last point refers to the argument of recall vs. precision, which we will touch upon later. As a high-level understanding, this means that the documents returned from the system need not be superfluous such that it costs more time and money inspecting them than the traditional methods.

The government guidelines also have a section regarding specialised technology as follows "If Electronic Documents are best accessed using technology which is not readily available to the party entitled to disclosure, and that party reasonably requires additional inspection facilities, the party making the disclosure shall co-operate in making available to the other party such reasonable additional inspection facilities as may be appropriate in order to afford inspection in accordance with rule 31.3." What this means is that if one party doesn't have access to the technology to complete e-disclosure, then another party must be open to supporting them to complete this task. As previously stated, this reiterates the fact that the system must be transparent, user friendly and repeat the same results with each input. This creates trust between the parties and makes a trustworthy system which completes the task successfully.

Having researched and learnt about the two tasks, e-discovery and e-disclosure, I have discovered that they are areas of high importance but also areas which have considerable costs of time and money. Creating a system which tackled these problems could be a great asset for the legal industry and further the push of technological innovation in law firms and other legal domains [6].

Due to the requirements of the tasks, and the scope of this project, we decided the system that we will aim to create is a legal question-answering system. It will be a system which will allow a user to input a set of data in the form of legal documents and the system, using NLP algorithms and techniques, will be able to answer questions inputted by the user and return the most relevant documents to the given question.

4 Objectives

This section will relay the aims and objectives of this project and why these are the specific objectives chosen. This allows us to have an aim for the end product and also relay what we

are striving to achieve in the scope of this project.

1.The first objective of this project is to create a system which makes e-disclosure and e-discovery, easier and more efficient. Having seen the number of costs these duties have, the system will allow to reduce them and create a more efficient and less tiresome process. Also, it creates a singularity in how the process is completed. One lawyer may not carry out the process, the same as another lawyer or paralegal. And these duties need to follow strict rules and regulations. A question-answering system allows us to do this because it doesn't have any pre-constructed knowledge or bias. We can program it to follow certain guidelines and it follow them every time. This means that it can be more reliable and we can trust that discovery and disclosure has been carried out under strict jurisdiction. We will achieve this by making a simple and straightforward model which completes the process in the same manner and can be tweaked to suit different needs in different situations.

2.Another objective of this system would be for it to be transparent and traceable while being secure and trusted. It has been repeatedly stated that the system has to be reliable in these areas as the industry it is supposed to apply to has extremely high standards regarding this. The data being inputted should only be sourced from the relevant parties and similarly, any outputs should only be transferred to any relevant parties. Security is of the highest concern and being able to trace the origin to the destination of the data, along with tracking any changes to the data achieves this requirement. In the system, we will show the span of the answer that pertains to the question and also which document it is being retrieved from, allowing the user to trace the answer and verify if its correct and from a file which is valid under the disclosure.

3.Finally, the system must be user friendly and achieve a good level of user experience, as the users of this system are unlikely to have experience in computer science. Therefore, the system must not be over complicated and have the user in mind when been implemented.

This means it should be easy to understand and the user should see the benefit of it. If the user believes it will take more time learning and using the system than doing it manually, we have effectively failed at creating a working system as the user itself cannot understand it.

Using these three aims, it was easier to conclude how the system should act and what it should achieve as an end product. It allows focusing the research and evaluating the progress.

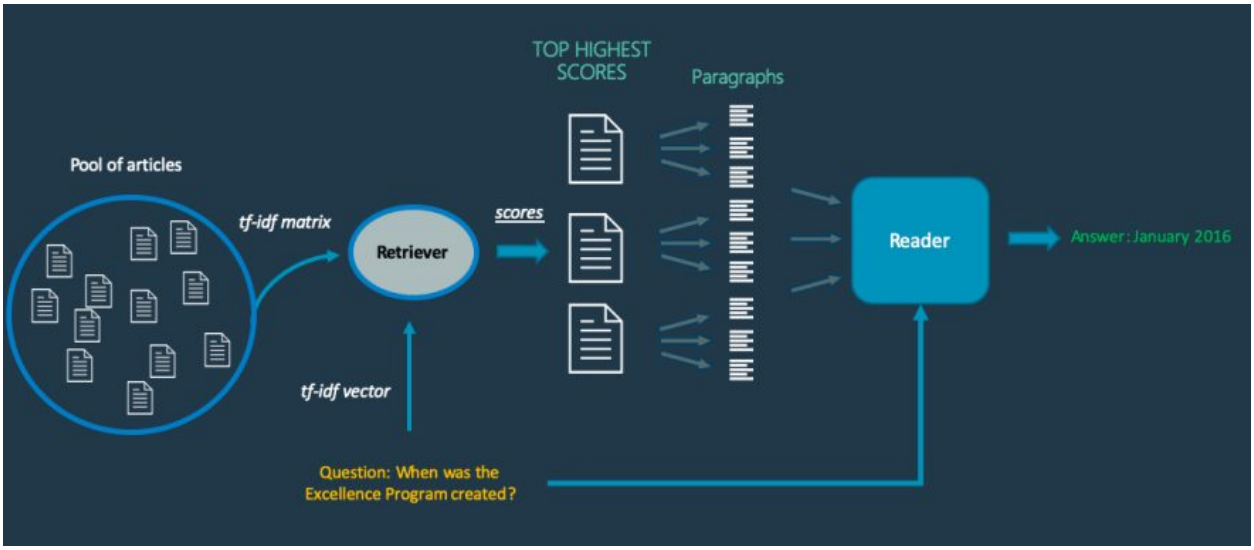
The question-answering system will first take in a large set of documents on which the user can ask the questions. The question will then be transformed and inputted into an information retrieval equation, along with the dataset of documents to find the most applicable documents to the question. The most relevant documents, as calculated by the equation, will be used as inputs for the pre-trained question answering the NLP model, which will allow us to find the answer spans and the expected right answer.

5 Design and Implementation

This section will cover the implementation, design and research behind the system which was created. It will detail the various steps which were taken to arrive at the final product.

5.1 Pipeline

First, we will describe the pipeline of the system as seen in the figure below. The pipeline describes the 'journey' the data takes through our system, and how our system calculates answer spans for each question. It is one-directional and we can trace it to the journey as the data is preserved at each step. The pipeline takes inputs of various types, such as the set of documents, the trained model and the question being asked.



5.2 Corpus

The pipeline begins with the dataset of documents, which is described as a corpus. A corpus is defined as the following:

"A corpus is a collection of pieces of language that are selected and ordered according to explicit linguistic criteria in order to be used as a sample of the language. Note that the non-committal word 'pieces' is used above, and not 'texts'. This is because of the question of sampling techniques used. If samples are to be all the same size, then they cannot all be texts. Most of them will be fragments of texts, arbitrarily detached from their contents. A computer corpus is a corpus which is encoded in a standardised and homogenous way for open-ended retrieval tasks. Its constituent pieces of language are documented as to their origins and provenance." [7]

Procuring a large dataset of legal texts is especially difficult due to a few reasons. First of all, in e-discovery, the data can come in many forms. They can be in the form of emails, contracts or even database entries. However, for our system, we decided to work with full-bodied documents such as only contracts or licenses. The problem then arose from finding

a suitable dataset to train our model and to ask questions. As a team, we couldn't find a dataset of legal texts with corresponding question-answer pairs to train our system, thus leaving us in a predicament. We could either create our own dataset with corresponding question-answer pairs or use a dataset which was not specific to the legal setting, however, was large and contained corresponding question-answer pairs.

For the first few weeks into our endeavour into this project, we completed a number of spikes to not necessarily try to make a product but understand the depth of the project and gain an understanding of all the moving parts. A spike is described as "A task aimed at answering a question or gathering information, rather than at producing shippable product. Sometimes a user story is generated that cannot be well estimated until the development team does some actual work to resolve a technical question or a design problem. The solution is to create a "spike," which is some work whose purpose is to provide the answer or solution." [8]

5.3 Spikes

For the spikes, we used simple text documents to test our methods. For instance, using Wikipedia articles as the input allowed us to create a generalised version of the project. Our task was simple and straightforward information retrieval task: retrieve the Wikipedia page and split the page into multiple paragraphs and then find the most common words to create a bag-of-words model. Using this model, we can find the most common words related to the input question to find the most relevant span, thus the answer to the question. It was quickly apparent that this task was not as simple as we first thought. The spike led us to learn many valuable lessons.

Firstly, we thought we could find the next paragraph using a new line character. This caused problems in the Wikipedia articles as they are usually split by sections. This lead to

single words being counted as paragraphs which skewed our outputs. Also, bullet points in lists were also being categorised as individual paragraphs when they were in fact supposed to be one singular span. From this, we learnt that the format of the individual documents will have great precedence and we will have to take into account a number of different types of formats and document structures.

5.4 Bag-of-Words Model

Once we have our list of paragraph spans, the next step is to create a bag-of-words model to find the most relevant paragraph by finding the common words between the paragraph and the question. To do this, however, we need to complete a few steps so that the words are formatted and parsable into our model. [10]

Our first step is transformation. What this means is, transforming the individual words into a standardised form. For example, this includes changing all the words to lowercase - the capitalisation has no effect on the importance of the word. Removing accents also allows us to then compare and parse information better. From these actions, we will have a set of words, which are easily comparable and parsable into our bag-of-words model.

The next step would be tokenisation. Tokenisation is defined as the following: "Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation." This allows us to break up the span into individual sentences or words so we can work with each token individually. This leads to the next step. [11]

Many words are the same basis, within a different context. For instance, loved, loving, loves are all forms of the verb "to love". To make our program work correctly, we needed to ensure that these words were recognised as the same word. Therefore, we need to apply the

method of normalisation. This consists of applying stemming and lemmatisation to words. Stemming algorithms work "by cutting off the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word". [12] This method doesn't always work, however, as common prefixes and suffixes exist in words which may not have a stem they originate from. On the other hand, Lemmetisation takes into consideration the morphological analysis of the words. While this is more effective, the computing power is considerably more as well, as we have to have a large dictionary of all the original words bases to compare to. This may not be expensive for a few documents, however scaling up to multiple documents in the thousands, the cost considerably goes up. Once this is done, we will see that the most common words seem to be connective words or words with a lack of relevancy to the question. These words include 'and', 'a', 'is' etc. The next stop of this process is dealing with this problem.

As stated in the previous paragraph, there is a problem of somewhat "useless" words dominating the frequency in our documents. Therefore, we must remove them to leave us with any operative words. This step is called stop word removal. This is as simple as importing a stop word list, which ranges from 5-10 words to 100-200 words and removing those from our word frequency table we have created. [13]

All the previous steps are carried out on the spans of text and also the question to find the span which has the most common words with the question, thus being the most relevant to answer it. While it is easy to find questions and answers for Wikipedia articles, the aim of this project is to be specific for the legal landscape. Therefore, we need to find a corpus which is more akin to the documents in that industry. However, as it is the legal industry, it was very difficult to find a suitable dataset with actual legal data and question-answer pairs on which we could train our model.

Therefore we decided as a team to use licenses in the early stages of the project. Licenses

are a good set for our corpus as they use many legal terms, have similar formats and are freely available. For actual testing purposes, we found a data set of multiple cases that we can use. It is large with many different case topics allowing us to really test the reliability of the system.

The choice of the corpus is extremely important as the more complicated and diverse the data set, the more reliable our system will be and more trustworthy it will be.

5.5 Term Frequency - Inverse Document Frequency

The term frequency information retrieval method can be extrapolated to a more formalised version. One such method is called TF-IDF, which means term frequency-inverse document frequency. It is a statistical measure of the relevance of a term to a document in comparison to the entire document set. For instance, words that appear many times across all documents will be ranked lower than words which appear many times in fewer documents as we take into account the inverse document frequency. This is helpful because words which are very popular such as 'if', 'and', and 'the' are not high on the TF-IDF scoreboard. [14]

Also, as this is a mathematical equation, we return numerical answers for each term, allowing us to create vectors for each. Word vectorisation, or word embeddings, are a method of giving terms in a document a numerical representation or visualisation. We can then use this for comparison, evaluations or further calculations. This is useful as machine learning algorithms mostly work with numerical data rather than raw strings or similar. Take the following quote for example: "Consider the sentence: "I am learning how word embeddings work". The words in this sentence are "learning", "embeddings" etc. From this, we can create a dictionary which is the list of all unique words in the sentence. In this case: ["I", "am", "learning", "how", "word", "embeddings", "work"]. A one-hot encoded vector representation of a word can be encoded in a way that 1 stands for the position where the

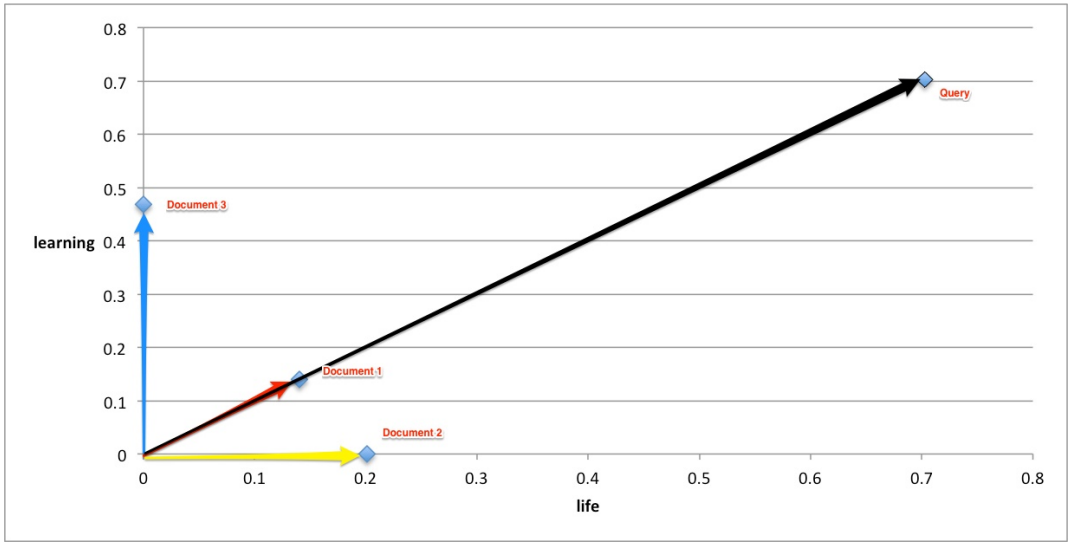
word exists and 0 everywhere else. For instance, the vector representation of "learning" is as follows: [0, 0, 1, 0, 0, 0, 0]. As a result, each word will have a unique dimension." This clearly represents how useful word vectorisation is in our situation, as we can pass it into a tf-idf matrix. We will also create a word vector for our question. Using these sets of vectors and matrices, we will be able to find the most relevant documents through the method of cosine similarity.

5.6 Cosine Similarity

Cosine similarity is as simple as measuring the distance between two distinct vectors. This is measured using the dot product of two vectors and reducing the distance between the two.

$$\textbf{similarity} = cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \, \|\mathbf{B}\|} = \frac{\sum \mathbf{A_i B_i}}{\sqrt{\sum \mathbf{A_i^2}} \sqrt{\sum \mathbf{B_i^2}}}$$

In our case, we do this between the terms in the documents and the question vector. This allows us to see the most relevant documents by finding the vectors with the smallest angle. We can choose the threshold for the angle, such that only the documents above this are chosen. Thus, we have now found the most relevant documents to passed into our model.



5.7 Bidirectional Encoder Representations from Transformers (BERT)

The model we have chosen to use is the Bidirectional Encoder Representations from Transformers (BERT) model developed by Google. Google developed this model to better understand users' searches and writing patterns. This is the model we will be using to find the spans of answers in our texts. [15]

One of the key aspects of the BERT model, as the name suggests, is that it is bidirectional. BERT's predictions work by masking a word in a sentence then looking at all the words before the masked word and then after the masked word (i.e. bidirectional) and trying to predict the hidden word from this.

BERT is trained on Wikipedia, however, it can be fine-tuned to be applicable to specific data sets. Sequences of words are processed by BERT wherein 15 percent of the sequence has been masked. It is then asked to predict the masked values in the inputted sequences. The steps can be described as follows:

1. Adding a classification layer on top of the encoder output.
2. Multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension.
3. Calculating the probability of each word in the vocabulary with softmax.

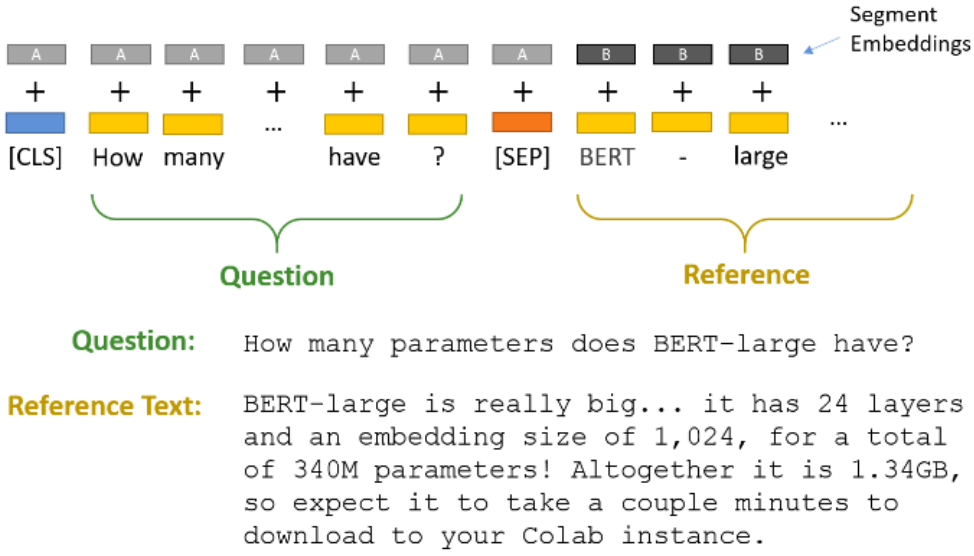
"The BERT loss function takes into consideration only the prediction of the masked values and ignores the prediction of the non-masked words. As a consequence, the moddel converges slower than directional models, a characteristic which is offset by its increased context awareness".

After this, BERT's next sentence prediction is trained. This can be defined as " the model receives pairs of sentences as input and learns to predict if the second sentence in

the pair is the subsequent sentence in the original document. During training, 50 percent of the inputs are a pair in which the second sentence is the subsequent sentence in the original document, while in the other 50 percent a random sentence from the corpus is chosen as the second sentence. The assumption is that the random sentence will be disconnected from the first sentence."

The sentences are differed using certain tokens as defined below.
A [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence. A sentence embedding indicating Sentence A or Sentence B is added to each token. Sentence embeddings are similar in concept to token embeddings with a vocabulary of 2.

A positional embedding is added to each token to indicate its position in the sequence. The concept and implementation of positional embedding are presented in the Transformer paper.



Then the following steps are taken to see if the first sentence is indeed connected to the previous sentence.

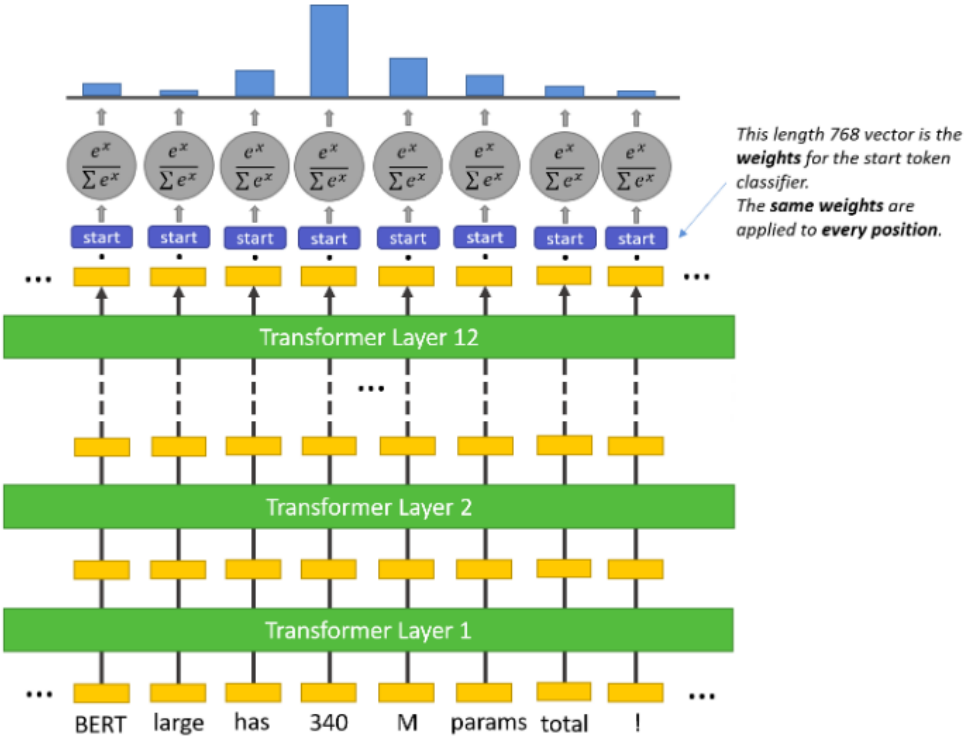
1.The entire input sequence goes through the Transformer model.

2. The output of the [CLS] token is transformed into a 21 shaped vector, using a simple classification layer (learned matrices of weights and biases).

3. Calculating the probability of IsNextSequence with softmax.

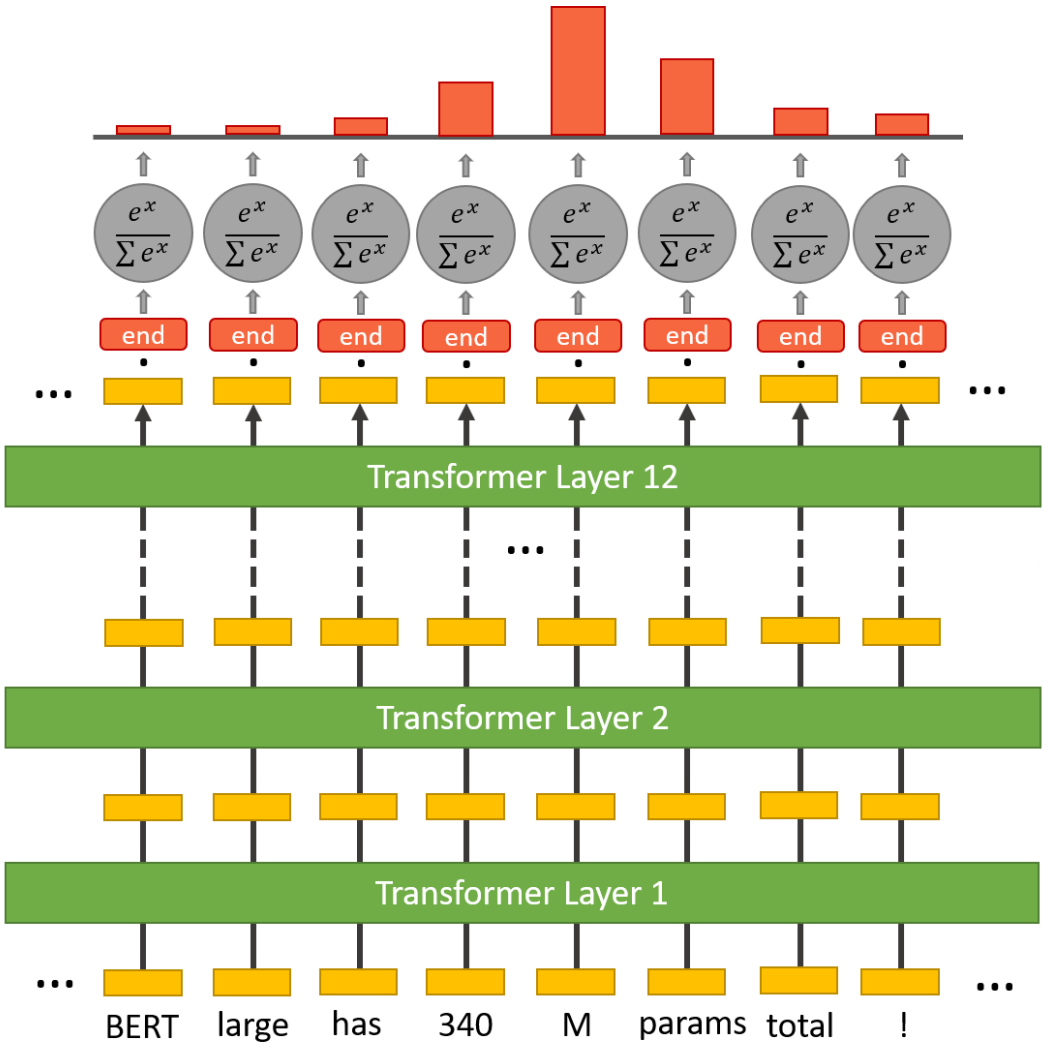
When training the BERT model, Masked LM and Next Sentence Prediction are trained together, with the goal of minimizing the combined loss function of the two strategies.

For question answering, BERT has been trained on SQuAD (Stanford Question Answering Dataset) [17]. Stanford Question Answering Dataset (SQuAD) is a new reading comprehension dataset, consisting of questions posed by crowd workers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage. It has 100,000+ question-answer pairs on 500+ articles and is significantly larger than previous reading comprehension datasets.



To answer a question, we feed into the model, the question and the data set as the inputs separated by tokens. The inputs are separated by a special [SEP] token, representing where a new segment begins. "Segment Embeddings" are also used to differentiate the question and the multiple spans of text, which in our case are the paragraphs from the documents. These are simply two embeddings (for segments "A" and "B") that BERT learned, and which it adds to the token embeddings before feeding them into the input layer.

BERT needs to highlight a "span" of a text containing the answer–this is represented as simply predicting which token marks the start of the answer, and which token marks the end.



Each token will have its final embedding fed into the start token classifier. The start token classifier only uses one set of weightings which are applied to every word. Then we can take the dot product of the output embeddings and the weightings, on which we apply a softmax activation to produce a probability distribution. The word with the highest probability is what we pick for the start of the span. Following this, the same occurs for the end token, with a separate weight vector. Thus, we return the span with the highest probability of including the answer in it. Then using NSP (next sentence prediction) as defined earlier, we can find the exact text which includes the answer.

6 Testing and Evaluation

To test our system, we as a team opted towards creating a gold standard for the licenses corpus and the cases data set as well. For question answering, there doesn't exist a set of questions and answers, therefore we created a document of questions and answers which under a reasonable doubt could test our system.

Using the set of questions and answers we created, we tested the system and noted the accuracy of the answers. We also tried to find which type of questions failed the system. Of the questions, 82% gave the right span and 76% gave the actual answer we were looking for. However, as these questions were created by ourselves, there was some bias involved and the questions may not be of the type which are expected from lawyers to be asking. Also, BERT seems to fail a lot more when there are negatives involved in the question. This is supported by Etinger's work regarding LM diagnosis and may be a reason where BERT model is unusable in our context and we may have to look towards an alternative such as AllenNLP. [16]

We can also look at the precision and recall of the system. Precision can be defined as "Precision = $\frac{TruePositive}{TruePositive+FalsePositive}$". What we infer from this is that precision is, the percentage of right answers which we have identified, to actually be correct. However, this also means that this does not take into account the correct answers, which were not identified whatsoever. Conversely, we also have recall to consider as a metric of evaluation. Recall is defined as "Recall = $\frac{TruePositive}{TruePositive+FalseNegative}$" as an equation. This means this includes all the right answers along with negatives which may be seen as right answers as well. However, with higher recall, the precision will fall and vice versa, with higher precision, recall will drop. Therefore there is an argument to which is more important, the accuracy of the answers which are actually returned, or the amount of answers there are such that no right answers are missed. To further assess these two metrics we can combine them into a single metric to return a score. This is called an F1 score and is described as "F1 = 2 x $\frac{Precision*Recall}{Precision+Recall}$". This score allows us to find a balance between the two metrics such that accuracy is as much as

possible. If it were not for the pandemic and related circumstances, delving into finding the highest F1 score would have been the next logical step in this project, along with finding a suitable fine-tuning method, such that we increase both precision and recall.

7 Reflection and Conclusion

There were many lessons learnt from completing this project. The most enjoyable part was researching different components of the system and understanding how they would all fit together. Trying to understand other peoples' work and implementing it into our own was also challenging and fulfilling.

The most interesting part of the project was the application of mathematics to natural language processing. As a mathematics student, these were the most captivating sections of the project and allowed me to have a better depth of understanding of natural language processing and machine learning. Seeing the different ways raw language data could be transformed into other forms for analysis was highly intriguing and we could see it applied to many other forms of data. If the project were to be done with hindsight, there would be a greater emphasis on client and user experience. We did not have the chance to meet any possible clients this time, which would allow us to see how usable our system would be.

It also showed me that natural language processing is not the field I want to go into as it. While the importance of the field is understated, the field is still growing and has a long way to go. If given the chance again, the project that would be chosen would be one with a lot more research already cemented with a lot more emphasis on mathematical data, rather than linguistic data. Nevertheless, this does not take away from the lessons this project has taught which we will carry into our careers.

Bibliography

[1] Bindman D. (2019). Law Society: Firms slow to adopt disruptive technology. https://www.legalfutures.co.uk/latest-news/law-society-firms-slow-to-adopt-disruptive-technology

[2] Heshmaty A. (2018). Legal tech in 2018: threats and opportunities. https://www.lawsociety.org.uk/news/blog/legal-tech-2018-threats-and-opportunities/

[3] D. W. Oard and W. Webber. (2013) Information Retrieval for E-Discovery. https://ediscovery.umiacs.umd.edu/pub/ow13fntir.pdf

[4] Larson A. (2018) Conducting Discovery in a Civil Lawsuit https://www.expertlaw.com/library/civil-litigation/conducting-discovery-civil-lawsuit

[5] Ministry of Justice. PRACTICE DIRECTION 31B – DISCLOSURE OF ELECTRONIC DOCUMENTS https://www.justice.gov.uk/courts/procedure-rules/civil/rules/part31/pd_part31b30.1

[6] Thompson S. (2016). Law firms slow to embrace 'new' technology due to a lack of desire and motivation to change https://www.lexisnexisinteraction.co.uk/blog/2016/01/07/law-firms-slow-to-embrace-new-technology/

[7] Corpus and computer corpus. http://www.ilc.cnr.it/EAGLES96/corpustyp/node5.html

[8] Agile Dictionary http://agiledictionary.com/209/spike/

[9] Chambers Students. (2017). Technology, innovation and law firms https://www.chambersstudent.co.uk/where-to-start/newsletter/technology-innovation-and-law-firms

[10] Orange3 Text Mining. (2019). https://orange3-text.readthedocs.io/en/latest/widgets/preprocesstext.html

[11] Stanford NLP. (2019). https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html

[12] Stanford NLP. (2019). https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

[13] IBM Knowledge Centre. Stop Word Removal. https://www.ibm.com/support/knowledgecenter/en/SS8NLW_12.0.0/com.ibm.discovery.es.ta.doc/iiys

[14] Stecanella B. (2019) What is TF-IDF?. https://monkeylearn.com/blog/what-is-tf-idf/

[15] Devlin J., Chang M. W., Lee K., and Toutanova K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/pdf/1810.04805v2.pdf

[16] Ettinger A. (2019). What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language modelshttps://arxiv.org/pdf/1907.13528.pdf

[17] Rajpurkar P., Zhang J., Lopyrev K., and Liang P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. https://arxiv.org/pdf/1606.05250.pdf