CS4099 SENIOR HONOURS PROJECT

Application of Transformer Models for Information Retrieval in the Legal Domain

University of St Andrews

James Zhang (190015412)

Supervised By Mark-Jan Nederhof

March 2024

Abstract

The legal domain is ever expanding, where laws, statutes, regulatory frameworks and legal precedents continue to evolve and expand at unprecedented rates. Consequently, the task of legal research has become increasingly complex and difficult to navigate. The ability to efficiently navigate pertinent legal information is indispensable, yet increasingly difficult for legal professionals. In its most basic form, legal research involves perusing thick paper volumes that typically line the book shelves of court rooms and libraries to find relevant information.

Information retrieval systems have played a pivotal role in the domain of legal research, serving as the cornerstone for accessing, analyzing, and synthesizing vast arrays of legal documents. These systems aim to not only enhance the efficiency of legal research but also significantly contribute to the accuracy and depth of legal analysis.

Traditional systems that are commonplace today rely on lexical methods and therefore lack consideration for the semantics and complex syntactical relations that exist in the natural language of legal text. This presents hard limitations in performance for a rather complex domain of language that is increasing in density and diversity. The goal of this project is to explore the application of a novel paradigm in natural language processing, transformer models, for the task of information retrieval. In particular, this project investigates the use of supervised and self-supervised training methods to fine-tune a Bidirectional Encoder Representations from Transformers (BERT) model to increase domain specificity and effectiveness for information retrieval of dense legal text.

Declaration

Declaration I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated.

The main text of this project report is 13361 words long, including project specification and plan.

In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

Contents

Chapter 1

Introduction

The characteristics of legal research demand meticulous and comprehensive analysis, requiring robust and specialized understanding of legal language, systems, and frameworks. Legal research spans from basic discovery in the case of litigation to doctrinal research for synthesis of new law. It is necessary for legal professionals to effectively discover and identify information relevant to a given topic, whether to provide context in forming arguments or develop legal strategies that underpin existing frameworks and precedents.

The legal domain is constantly expanding. Laws, statutes, regulatory frameworks and legal precedents continue to evolve and expand at unprecedented rates and vary in format and language depending on many factors such as time, jurisdiction, or type of document. Consequently, the task of legal research has become increasingly complex and difficult to navigate, even for experts.

Legal language additionally presents unique challenges for comprehension. As a specialized domain, legal text features uniquely complex sentence and document structures, specialized jargon, and ambiguities requiring expert knowledge to navigate. The ever-expanding volume of legal information, alongside the nuanced and unique nature of legal language, underscores the critical need for advanced information retrieval methods for effective legal research.

Traditional information retrieval systems, which are widely used today, primarily rely on lexical methods for search operations. Consequently, they fall short in accounting for the nuanced semantics and complex syntactical relationships characteristic of legal texts. Lexical strategies hinge on the statistics of exact keyword appearances and their matching, thus placing more responsibility on the user of the system to carefully form effective queries to locate and retrieve information.

Innovations in Natural Language Processing (NLP) present a new perspective for approaching the task of information retrieval. These advancements enable more sophisticated analysis of text, moving beyond lexical methods to capture and understand context, semantic relationship, and nuanced meanings present in legal text and queries. By leveraging techniques such as vector embedding and machine learning for semantic analysis, information retrieval systems can pinpoint relevant information with much greater precision and recall, making processes such as legal discovery and doctrinal research more thorough and effective.

The complexities of legal research and requirements for expert knowledge in the legal domain means that data collection and labelling for NLP solutions is both costly and cumbersome, leading to scarce training data. The goal of this project is to investigate the application of Transformer models [2] for legal text analysis and information retrieval. In doing so, we explore an array of training approaches such as domain-specific further pre-training and multi-task fine-tuning to increase performance and domain specificity for the task of information retrieval without relying on an abundance of expert labelled data.

1.1 Project Objectives

The objectives of this project can be grouped into an exploratory research component and a software engineering component. The aim of the research component is to develop methodologies to fine-tune a transformer model for the task of information retrieval in the legal domain. On the software engineering side, the imperative is to use the resulting trained model to develop a usable application for performing semantic searches over a corpus of legal text. Accordingly, the objectives for this project are as follows:

1.1.1 Primary Objectives

1. Dataset Selection & Collection : Select a number of corpora and datasets suitable for the selected fine-tuning tasks and to serve as the corpus to perform information retrieval over. This objective includes implementing suitable data pre-processing steps.

2. Model Selection and Training Methodology : Select a suitable pre-trained transformer model for the domain and identify downstream tasks to use for fine-tuning the model for the legal domain and the task of information retrieval. This involves creating a number of fine-tuned models using a number of various fine-tuning strategies.

3. Bench marking & Evaluation : Benchmark and evaluate the performance of each fine-tuned model against each of the selected downstream tasks for fine-tuning as well as a the primary task of information retrieval.

1.1.2 Secondary Objectives

1. Implement Semantic Search: Utilize the fine-tuned model and an appropriate corpus for information retrieval. Develop a semantic search pipeline to process queries and deliver relevant results.

2. User Interface : Develop a user application around the core functionality of the fine-tuned model for information retrieval. This involves implementing a graphical user interface and backend surrounding the core information retrieval functionality as a proof of concept to visualize the search functionality and to explore common pipelines used for information retrieval

1.2 Software Engineering Process

The development methodology of this project was based on an Agile iterative and incremental development approach. Due to the exploratory nature of this project, it was essential to use a development framework that can adapt to changing scopes, requirements, and unforeseen obstacles. As a result, I stayed away from more traditional waterfall methods which involve approaching development in stringent pre-defined stages, moving between each stage from beginning to end in one pass.

A large component of this project is machine learning, and in particular, the fine-tuning of large language models. Machine learning naturally requires a lot of experimentation and exploration with various models, features, parameters, and data selection and processing techniques. The process involves continuous testing, evaluation, and refinement based on performance metrics and feedback. So naturally, I found an Agile approach to be most suitable for this component of the project.

In regards to the iterative nature of my development process, I found it best to start with a high-level basic implementation, often leveraging pre-built functions and tools, and over time, iteratively re-work various components at a lower level. With this development life-cycle, each iteration produces a working product, although not feature-complete, and refines the product, adapting to changes and nuances while incorporating feedback. By starting with a high-level basic implementation, this approach ensures a functional product early on that can be tested and evaluated to produce feedback useful for guiding further development and identification of areas for improvements.

1.2.1 Languages & Technologies

All fine-tuning procedures were written in Python. This choice was motivated by Python's rich ecosystem of libraries and frameworks that support machine learning tasks. In particular, I am utilizing the HuggingFace `Transformers` library, which provides a massive library of Transformer model architectures including BERT and GPT base models and variants. The library offers a unified API for accessing each type of model, making it ideal for this project which necessitates switching between several models for experimentation [46]. The `Transformers` Library also comes with massive open-source repository of pre-trained and fine-tuned models, providing more opportunities for bench marking different pre-trained and fine-tuned models against each other. The library also provides a number task specific heads for each Transformer architecture. For example, the BERT model has an array of options for masked language modelling, sequence classification, and generative question & answering task-specific output layers.

The PyTorch framework is used for handling the lower level logic and operations of the training routines. This was a natural choice considering that the `Transformers` library is built on-top of PyTorch [55]. In fact, each of the task-specific layers offered by the Transformer library are sub-classes of the core PyTorch `nn.Module` class, inheriting its core functionalities such as `forward` and `backward` methods for making predictions and back-propagating loss gradients respectively.

Additionally, the `nltk` library is used for handling lexical operations such as the construction and utilization of TF-IDF for lexical search and similarity scoring. This library is also used to implement a basic trigram language model for bench-marking the masked language modelling task (more in section 3.2).

The user-interface consists primarily of a back-end server and a graphical client. The server is written in Python using the Flask framework for implementing a REST API with the client. The client is implemented using the ReactJS JavaScript framework.

1.3 Ethics

This project involves the collection of legal documents and text including case holdings, case law documents, and legal statutes. All data is publicly available and open source, originating from public entities such as courts. However, this type of data contains the names of individuals, potentially in a sensitive context such as in court cases. The content of this data is not the focus on this project, but rather the use of legal language is. The primary use of this data is for fine-tuning and evaluating large language models as a proof of concept. There is no additional risk introduced with this project in regards to management of sensitive data – no additional inferences will be made regarding the content of the data nor the generation of any derivative content. The usage of this data is only for research purposes and will not be used for any commercial purposes nor distributed.

An ethical application was submitted to the School of Computer Science Ethics Committee and was approved.

Chapter 2

Context Survey

In this section, we explore the background and nature of the legal domain and information retrieval so as to provide context into what motivates this project and the challenges that must be overcome. Additionally we review the literature relevant to this project, including past experiments in the same space and approaches for the same task.

2.1 Characteristics and Challenges of the Legal Domain

The legal domain, like many other specialized domains, has distinct characteristics compared to everyday language. It can be quite difficult to formally define what legal language is precisely or what makes a language *legal*. The legal domain can be characterized as a sub-language [64, 22]. While grounded in everyday general language, the legal domain features unique characteristics such as jargon or document and sentence structures. The degree to which a legal text diverges from everyday general text also varies. While some texts are meant to effectively and efficiently convey information between specialists within the field, others are intended to be understandable to the layman [21].

There are a vast number of characteristics of the legal domain that are important to consider when approaching the task of information retrieval and natural language understanding. Addressing all characteristics is quite complex due to the size and heterogeneity of the legal domain. The following are the key characteristics this project attempts to address – while far from exhaustive, these characteristics paint a picture of the complexities of legal language that motivate the usage of NLP approaches for information retrieval:

1. Legal Terminology : The lexicon of legal language is marked by a complex interplay of synonymy, ambiguity, and polysemy, where terms can possess extremely precise or vague definitions [52]. This includes specialized jargon unique to the legal field or instances where common words contain distinct legal meanings divergent from their everyday use.

2. Structure : Legal documents are known for their meticulous structure, adhering to rigid organizational patterns at both the macro (document) and micro (sentence) levels. This structured nature ensures clarity and precision, crucial for legal interpretation.

3. Document Size : legal documents tend to be quite large and dense, which reflects the rich and precise nature of the language and the necessity to minimize ambiguity.

4. Multi-jurisdictional : The characteristics of legal text and documents vary from jurisdiction to jurisdiction. This variability highlights the need for nuanced understanding of various frameworks and terminologies when dealing with multi-jurisdictional or international law.

5. Temporal Aspects : The legal domain is subject to continuous evolution, with legal texts expanding and adapting over time. This dynamic nature complicates the tracking of legal developments and requires constant vigilance to changes in legal vocabulary and interpretations.

2.1.1 Legal Terminology

Legal language has a rich and very specific vocabulary which complicates natural language understanding. For one it contains unique jargon with usage that is difficult to capture using a general corpus [22, 5]. For example, Latin terms such as *res judicata*, meaning "a matter that has been adjudicated by a competent court and therefore may not be pursued further by the same parties" [16], exist almost exclusively in the legal domain to efficiently convey legal concepts. Hemel (2022) [26] refers to this category of vocabulary as "legalogisms", denoting legal terms that have no colloquial correspondent.

The issue of polysemy is rather prevalent in the understanding of legal text as well [68]. This arises when a word or phrase has a different meaning in legal language compared to non-legal language. For example, in the United States legal domain, the word "take" has the definition of "significant habitat modification or degradation where it actually kills or injures wildlife", which was defined by the court case *Babbitt v. Sweet Home Chapter of Communities for a Great Oregon 515 U.S. 687* however this definition does appear in any English dictionary [26]. Another example could be the term "consideration" which in the legal domain is a key ingredient for an enforceable contract [11] serving as "promise, performance, or forbearance bargained by a promisor in exchange for their promise" within a contract [10].

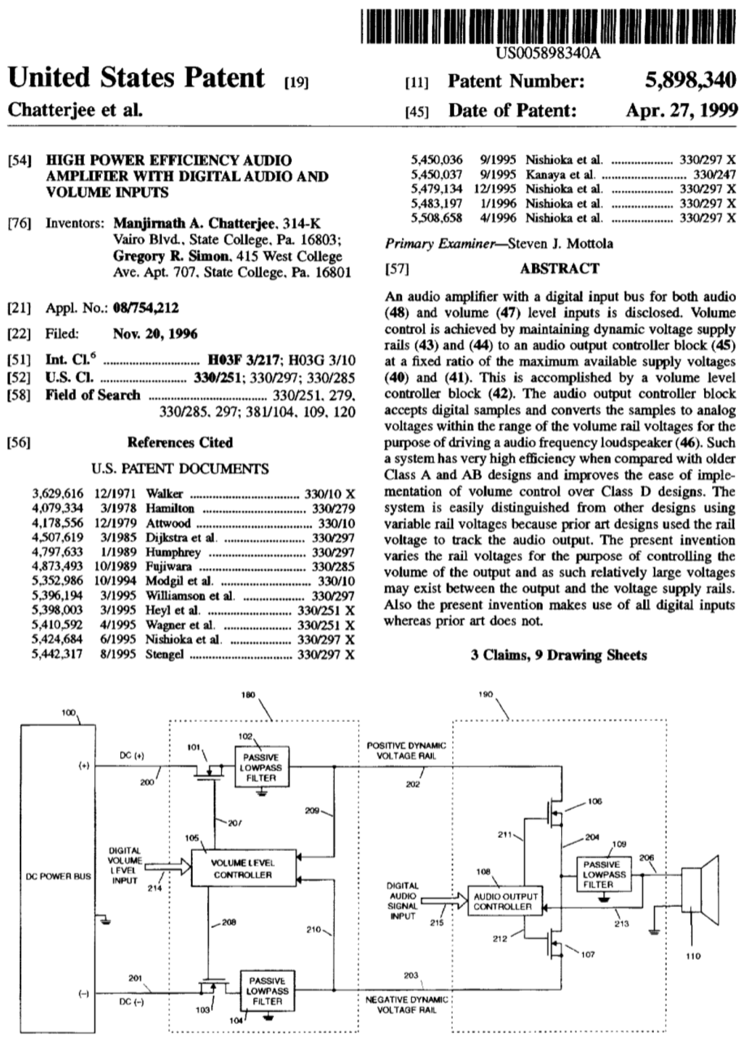
Moreover, in the legal domain, definitions constantly evolve. Much of legal action, including court cases, fundamentally revolves around the nuanced debate over the definitions of specific terms, where the interpretation can significantly impact the outcome. For example, the definition of *privacy* has undergone much change since its conception, primarily due to societal and technological changes. Historically, the concept of privacy was primarily associated with physical intrusion of one's private space [42]. Overtime, especially with the advent of information technologies, the definition has expanded. For instance, in 1890, the article *The Right to Privacy* [69], which was massively influential on the formation United States Tort law [42], provided a new definition of the word *privacy* as the general "right to be left alone", taking into consideration the introduction of instantaneous photographs, which had become common for newspapers [69, 42]. In particular, the article argues that the right to privacy goes beyond mere physical intrusion but includes any disclosure of private facts, thoughts, and even emotions [42, 69].

Legal language also varies across jurisdictions, adding complexity and ambiguity in its understanding. Different jurisdiction may contain different definitions for the same term, or potentially use different terms to describe the same concepts. For instance, the term *manslaughter* is used in the United States legal system to refer to any unlawful killing of a human being without malice [1]. However, in Scotland, the equivalent term is *culpable* *homicide* [14].

These ambiguities and differences between legal and non-legal language demands thorough and specialized understanding of the domain in order to navigate and decipher. As such, information retrieval systems must be created and trained in such a manner to capture these complexities and to facilitate accurate and precise access to legal resources.

2.1.2 Legal Documents

The complex structure and size of legal documents further complicates the task of understanding and information retrieval. This comes from the need to ensure precision, unambiguity, and comprehensiveness in legal texts such as contracts, court rulings, case law, legislation, and regulatory documents to name a few [66]. Legal documents are additionally not designed with machine readability in mind [52]. Take for instance patent documents, which include structured claims regarding the legal boundaries of an invention's protection, often structured in a hierarchical format, along with detailed descriptions and illustrations explaining the functionality of the invention [73].



US005898340A

United States Patent [19]
Chatterjee et al.

[11] **Patent Number: 5,898,340**
[45] **Date of Patent: Apr. 27, 1999**

[54] **HIGH POWER EFFICIENCY AUDIO AMPLIFIER WITH DIGITAL AUDIO AND VOLUME INPUTS**

[76] Inventors: **Manjirnath A. Chatterjee**, 314-K Vairo Blvd., State College, Pa. 16803; **Gregory R. Simon**, 415 West College Ave. Apt. 707, State College, Pa. 16801

[21] Appl. No.: **08/754,212**

[22] Filed: **Nov. 20, 1996**

[51] **Int. Cl.**^{6} **H03F 3/217**; H03G 3/10
[52] **U.S. Cl.** **330/251**; 330/297; 330/285
[58] **Field of Search** 330/251, 279, 330/285, 297; 381/104, 109, 120

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,629,616	12/1971	Walker	330/10 X
4,079,334	3/1978	Hamilton	330/279
4,178,556	12/1979	Attwood	330/10
4,507,619	3/1985	Dijkstra et al.	330/297
4,797,633	1/1989	Humphrey	330/297
4,873,493	10/1989	Fujiwara	330/285
5,352,986	10/1994	Modgil et al.	330/10
5,396,194	3/1995	Williamson et al.	330/297
5,398,003	3/1995	Heyl et al.	330/251 X
5,410,592	4/1995	Wagner et al.	330/251 X
5,424,684	6/1995	Nishioka et al.	330/297 X
5,442,317	8/1995	Stengel	330/297 X
5,450,036	9/1995	Nishioka et al.	330/297 X
5,450,037	9/1995	Kanaya et al.	330/247
5,479,134	12/1995	Nishioka et al.	330/297 X
5,483,197	1/1996	Nishioka et al.	330/297 X
5,508,658	4/1996	Nishioka et al.	330/297 X

Primary Examiner—Steven J. Mottola

[57] **ABSTRACT**

An audio amplifier with a digital input bus for both audio (**48**) and volume (**47**) level inputs is disclosed. Volume control is achieved by maintaining dynamic voltage supply rails (**43**) and (**44**) to an audio output controller block (**45**) at a fixed ratio of the maximum available supply voltages (**40**) and (**41**). This is accomplished by a volume level controller block (**42**). The audio output controller block accepts digital samples and converts the samples to analog voltages within the range of the volume rail voltages for the purpose of driving a audio frequency loudspeaker (**46**). Such a system has very high efficiency when compared with older Class A and AB designs and improves the ease of implementation of volume control over Class D designs. The system is easily distinguished from other designs using variable rail voltages because prior art designs used the rail voltage to track the audio output. The present invention varies the rail voltages for the purpose of controlling the volume of the output and as such relatively large voltages may exist between the output and the voltage supply rails. Also the present invention makes use of all digital inputs whereas prior art does not.

3 Claims, 9 Drawing Sheets

Figure 2.1: Example of A United States Patent [7]

The inherently long and dense nature of legal documents poses a significant challenge in machine understanding as well. The extensive length of legal documents increases the difficulty in finding efficient machine representations that capture the entire semantic understanding of texts [66]. Algorithms must maintain a high level of accuracy across large volumes of text, often containing multiple interrelated legal arguments and references [66, 59]. This becomes an issue for computational resources, thereby complicating efficient and accurate information retrieval and analysis [22, 59].

2.2 Transformer Model

Introduced in the 2017 paper "Attention is All You Need" by Vaswani et al. [2], transformer models were created to address the limitations of the Recurrent Neural Networks (RNN), which were the leading natural language processing paradigm. In particular, they introduce a different approach for processing long range dependencies that exist in natural word sequences [2].

Recurrent neural networks are designed to handle sequential data, such as text or time series data [25]. They function by maintaining a hidden state to effectively carry and represent information across the processing of a sequence. The architecture maintains this hidden state through a process of weight adjustment, as is standard for neural networks. At each step along a sequence, the hidden state is updated based on the current input/event and the previous hidden state [25].

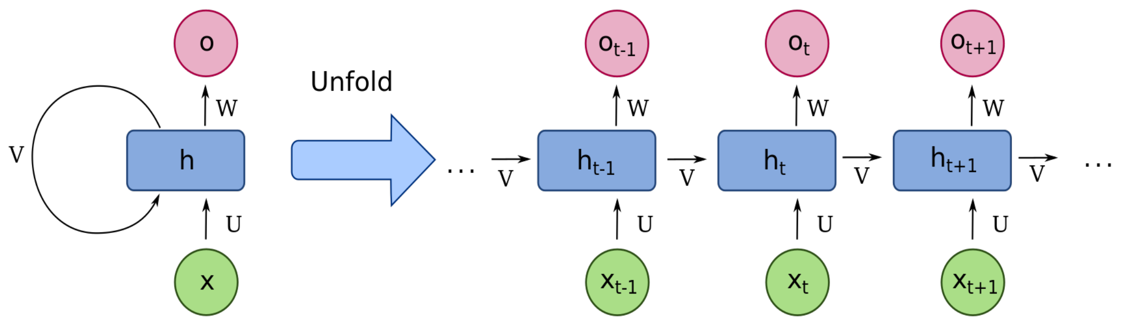


Figure 2.2: Basic RNN Architecture [19]

In Fig 2.2, `x` denotes the input, `h` denotes the hidden state, and `O` denotes the output. We can see that at each position along the input sequence, the output is a function of the input and hidden state at that position. At each step i along the sequence, the hidden state h_i is a function of the hidden state at the previous position h_{i-1} and the current input x_i. The goal for this mechanism is to retain all information processed from the beginning of a sequence, which is necessary to capture context for making language predictions. In the context of language processing, this mechanism allows for the generation of context-aware word embedding since embeddings within sequences can be generated considering information earlier in the sequence.

This architecture is sequential in nature [2]. This poses a crucial limitation in handling long-range dependencies in sequential data like natural language or time-series, where distant elements may be interdependent [2]. The farther apart these elements are, the harder it becomes to transmit information between them using the hidden state mechanism, crucial for understanding and modeling the data accurately [60]. This also causes issues for the computation and propagation of gradients during RNN training. At each position of a sequence, the model makes a prediction for which a notion of loss can be computed. Like any other neural network architecture, to minimize the loss, the gradient of the loss needs to be calculated and propagated through the neural network layers to adjust prediction weights. However, as each prediction relies on prior ones via the hidden state mechanism, gradients must be back propagated across the entire sequence, across each recurrent connection. As a result, gradients can multiply or decay exponentially as they travel back along a sequence, known as the exploding and vanish gradient problems respectively [60, 2]. This essentially leads to unstable weight optimization due to the inability to accurately propagate information along sequences [60].

Transformer models [2] are a fundamentally new architecture that introduces the use of a parallel mechanism called self-attention. This new approach allows for modelling of sequences and capturing of dependencies between events regardless of distance in the sequence [2]. The process enables the creation of context-aware embeddings in parallel, offering significant improvements over traditional sequential methods [2].

Self-attention operates by computing a score for each pair of positions in an input sequence [2]. This is done to reflect how much focus/weight should be given to other parts (tokens) of the sequence when encoding the token at a particular position [2]. These scores are obtained through a dot product of an initial embedding of a query (the position being encoded) with initial embeddings for all keys (other positions in the sequence), followed by a soft-max operation to normalize the scores to ensure they are normalized probabilities [2]. Next these weights are multiplied against the initial embeddings for all words in the sentence, effectively re-weighing all tokens in the sequence with respect to the token in the query position [58, 2]. This process is repeated for every word in the sequence, so each word of the sequence effectively obtains some context from every other word in the sequence. This process is done in parallel for all positions in the sequence as the computation for each position is independent from each other [2].

Interestingly, this process does not take into consideration any notion of proximity between tokens nor the number of tokens in a sequence, unlike other approaches such as RNNs [2]. The self-attention mechanism effectively allows the model to focus on different parts of the input sequence when making predictions, enabling it to capture dependencies between distant elements in the sequence without constraints of sequential processing [2]. Moreover, the Transformer model uses multiple heads of self-attention in parallel, called multi-head attention, to capture different types of dependencies from different representation sub-spaces at different positions [2].

A crucial aspect that this mechanism enables is context-dependent embeddings. Unlike static embedding approaches such as *GloVe* or *word2vec*, with transformer models, word embeddings are dynamic and differ depending on their context [2]. This enables the model to disambiguate words depending on its context [29, 2]. This capability is especially vital in the legal domain, which is rife with ambiguity. Words can have different meanings based on contextual factors such as legal jurisdiction or may require differentiation between their legal and colloquial definitions.

2.3 BERT Model

The Bi-directional Encoder Representations From Transformers (BERT) [17] model has proven to be very effective for solving many of the common tasks within the field of natural language processing from sentiment analysis to language modelling tasks. It is a specific implementation of Transformer-based large language model, designed for powerful context aware encoding, that introduces a unique and versatile architecture, applicable for a large variety of language tasks, including information retrieval. In fact, BERT helps power Google's state-of-the-art search engine, empowering semantic searches that captures the intent behind queries [47].

The core BERT model is a base encoder transformer model which is used for producing rich context aware embeddings for tokens [17]. In particular, the core model consists of multiple stacked encoder layers, where each layer primarily consists of multi-head self attention mechanism (as defined by the Transformer architecture), and a position-wise fully connected feed forward network [2]. Through the multi-head self-attention component, the core model is able to produce rich context aware embeddings for each token in an input sequence that weights the importance of all other words in the sequence. Since the self-attention mechanism doesn't process tokens sequentially, in order to capture the notion of order between tokens, additional positional information is encoded into the tokens [17].

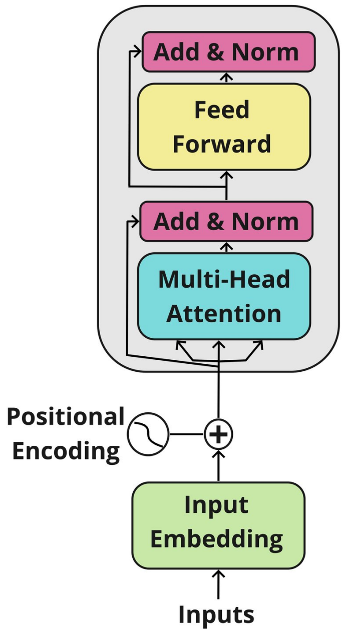


Figure 2.3: Transformer Encoder Block Architecture [2]

The power behind BERT comes from its pre-training approach and task selection, which compliments the design of the Transformer attention mechanism. In particular, BERT utilizes two self-supervised tasks : Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). MLM pre-training is a crucial task because it enables the learning of deep bi-directional context-rich representations of words [17, 58]. The tasks consists of masking (replace with a `[MASK]` token) some percentage of tokens in an input sequence, where the task is to predict the masked token. MLM encourages models to understand both the syntactic structure and semantic relationships within the text [29]. This approach not only leverages the full context of the input sequence, aligning perfectly with the Transformer's self-attention mechanism, but also enhances data efficiency and model adaptability for a wide range of downstream tasks [17, 29].

Next sentence prediction pre-training is performed to enhance the model's understanding of relationships between sentences, or generally larger inputs, which is a task not addressed by MLM [29, 17]. The task involves predicting whether a second sentence follows a first sentence in a document, encouraging the model to learn contextual relationships and coherence across full sentences or spans of tokens [29, 17]. In particular, BERT makes uses of a sentence-level token `[CLS]` used to encode the overall meaning of sentences. NSP is designed to optimize the model's generation of these `[CLS]` tokens [29]. This was done to support down stream tasks such as question and answering or text summarization which require capturing semantic information of spans of tokens in a single representation [17]. In the context of information retrieval and capturing relationships between two sequences (i.e. a query and result sequence), BERT natively supports encoding two sequences together, separated using a `[SEP]` token [17, 29]. This enables the self-attention mechanism to attend to tokens across both sequences in effort to produce an optimal [CLS] token embedding that can capture the similarity or some other form of classification of the combined sequence, allowing for richer and more nuanced analysis [17, 29].

2.3.1 Fine-Tuning BERT

A key feature of the BERT architecture is that due to its robust core encoder model, the pre-trained BERT model can be fine-tuned to specific tasks with just one additional output layer [17]. This allows the core encoder model to be leveraged for a variety of downstream tasks without substantial architectural modifications. The primary benefit of this feature is transfer learning [29]. However, task-specific fine-tuning can be leveraged to not only train an additional lightweight task-specific output layer, but also adjust the parameters of the core encoder model [29]. For this reason, fine-tuning is not only a tool for improving task-specific performance, but additionally to influence the way the core model produces its embeddings to understand and encode language. For example, Rogers et al. (2020) [58] demonstrates how the linguistic information across the layers of BERT changes during fine-tuning. In particular, the study investigates how the encoding of different types of linguistic information (from syntactical to semantic) exist in a hierarchical manner over the layers of BERT and the extent to which fine-tuning affects each of the layers [58]. The study finds that fine-tuning not only affects the higher encoding layers responsible for semantic understanding, but also lower level layers responsible for more syntactical linguistic features such as grammar [58]. Additionally, Zhao et al. (2020) [75] provides a comprehensive investigation into how fine-tuning tasks not only adjust weights of outer task-specific classification layers, but additionally changes the core functionality and sensitivity of the attention-head mechanism using the method of *attention head probing* [9][75]. Using the task of negation scoping (identifying what region of a sequence a negation cue/term applies to), the study found that the sensitivity of the BERT model attention heads increased with respect to negation cues and scopes after fine-tuning[75].

Many works have shown that fine-tuning BERT on specific domains of language can both improve performance for the specific fine-tuning task, and also increase domain specificity [24, 62, 49, 5]. This is sometimes referred as *further pre-training* [24, 62]. This is particularly important for adapting a model to complex domain-specific tasks which lack expert labeled data [23]. With this scheme, to fine-tune a model for a specific task in a specific domain, additional pre-training can be utilized to address the deficiencies in annotated task-specific training data. A common approach as exhibited in [24, 62, 49, 5] is to repeat the pre-training tasks used by BERT, in particular MLM, using an extensive in-domain dataset. This constitutes a self-supervised training routine that helps address sparse data problems. Interestingly, Chalkidis et al. (2020) [5] finds that the use of in-domain fine-tuning (further pre-training) obtains comparable performance for domain-specific tasks as training a model using domain specific data from scratch despite the former method being more lightweight and efficient to carryout [5]. These investigations underscore the utility in using fine-tuning methods to improve the performance of models despite scarce task and domain specific data.

2.4 Multi Objective Fine-Tuning

While the fine-tuning of BERT has been thoroughly explored, most efforts focus on single-task fine-tuning. For example, Zheng et al. (2021) [76] presents a standard method for single-task fine-tuning of BERT for the task of multiple choice answering over court case holdings. This is essentially a binary classification task where given two sequences, a court case citing context and a case holding, the task is to determine whether the case holding semantically follows the citing context [76]. To fine-tune the BERT model, Zheng et al. (2021) [76] attaches a sequence classification layer over the core BERT model, and trains the model using a dataset of case holdings (where samples are a citing context along with either a correct or incorrect corresponding case holding) [76].

An alternative approach to fine-tuning that is less discussed is multi-task fine-tuning. The primary idea is to simultaneously fine-tune or train the same core BERT model using multiple tasks. In particular, each task shares the same core BERT layers for encoding but have separate final task-specific classification layers [40, 62]. This has several potential benefits as outlined by Liu et al. (2019) [40]. Firstly, fine-tuning procedures that employ multiple tasks can make use of a vast amount of relevant domain-specific data, instead of being confined to potentially scarce data specific to a single task which may be rather obscure [40]. Secondly, this approach provides somewhat of a regularization effect, allowing the core BERT model to learn more robust and general representations of in domain text – essentially alleviates over fitting to a specific task and improves overall domain-specificity, providing benefits for any in-domain task [40].

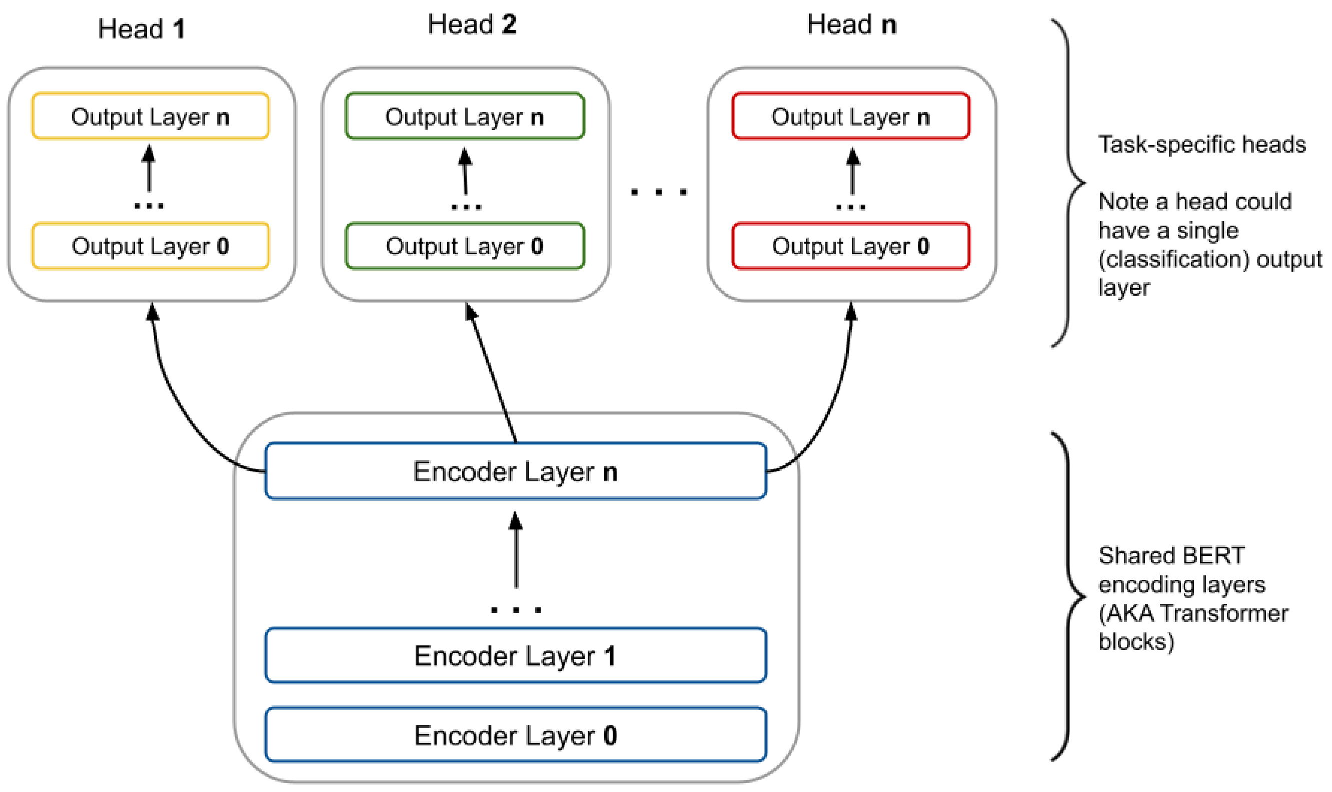


Figure 2.4: Multi-Task Bert Architecture

There are two paradigms for multi-task fine-tuning : sequential training and parallel training [38]. There are many approaches for each paradigm. For sequential training, one could individually fine-tune the same model for a number of tasks, one after the other. Alternatively, the routine could switch between tasks at each epoch or batch. As the name suggests, parallel training involves training each task simultaneously with each batch. For each training input, the model makes parameter adjustments in parallel based on the loss for each prediction for each task. Furthermore, with either paradigm, configurations can be made to produce a weighted sum or linear combination of sorts as to weight the effect of each task differently [34].

Each multi-task fine-tuning approach has its own set of challenges and drawbacks. For example, a common phenomenon called *negative transfer* can occur if the different tasks seek conflicting representations – this is to say that the loss gradients computed for each task prediction point in different directions [74, 34]. The common approach for dealing with divergent gradients is to utilize some sort of aggregation technique such as averaging [34]. Another important phenomenon that is more common for sequential multi-task approaches is *catastrophic forgetting* [45]. This occurs when a neural network, after being trained on one task and then trained on another, retains most of the information related to the second task while significantly losing the information learned from the first task – thereby "forgetting" the knowledge previously acquired [45, 62].

2.5 Information Retrieval

Information Retrieval (IR) is the process of obtaining relevant information or resources in response to a specific query. There are essentially three primary mechanisms required for

IR:

1. Indexing : creation of an efficient structure that allows for efficient queries to be performed over. This typically involves extraction and representation of key terms, features, and information generally to their locations (resources) in the index.
2. Query Processing : understanding and interpreting a user's search query, which can involve a number of techniques to ingest or understand the query. This can involve NLP techniques or lexical methods.
3. Search & Ranking : Searching of the index to find matches for a query and ranking of these matches based on some notion of relevance. Relevance can be based on lexical measures or more advanced semantic measures

2.5.1 Background

Generally IR systems can be grouped into two paradigms for search : lexical search and semantic search. Lexical information retrieval is based upon exact matching of words or phrases in a query with those in the information database. In a way, this approach operates at a more surface level, relying directly on the presence of specific terms within text to determine a notion of relevance or similarity. At the most basic level, lexical search utilises exact query matching. More advanced methods can utilise statistics to capture occurrence and distribution of terms within a corpus. These methods go beyond simple query matching, incorporating statistical measures to evaluate how informative or significant a term is across a corpus.

A common implementation is Term Frequency Inverse Document Frequency (TF-IDF), which aims to give a numerical statistic measuring how important a word is to a document within a larger corpus of documents [61, 56]. This therefore provides a notion of how important a document is to a query based on their word content [61, 56]. The TF-IDF method functions by computing Term Frequency (TF) in a document, and Inverse Document Frequency (IDF) which captures a notion of how often terms occur across the whole document corpus [56]. The two components are computed as follows

$$\begin{aligned} TF &= \frac{\text{\# of occurrences of a term T}}{\text{total \# of terms in document D}} \\ IDF &= \log \frac{\text{\# of documents in corpus}}{\text{\# of documents in corpus containing term T}} \\ \text{TF-IDF} &= TF \times IDF \end{aligned} \tag{2.1}$$

A number of similar statistic-based lexical methods exist including Bag-Of-Words [48] or BM-25 [54]. Lexical methods are quite widespread as a means for IR. They are rather lightweight to setup and computationally cheap. Consequently, many industry standard IR systems in the legal domain have been lexical in nature.

One of the first legal IR systems is LexisNexus, which emerged in the 1970s, aimed to assist legal research in the United States [27]. Shortly after, numerous other tools were developed, with Westlaw being the most prominent [71], which served as an IR system over case law documents, federal statutes, administrative codes, as well as news and magazine articles [71]. These two popular information retrieval systems are traditionally lexical and capitalize on boolean search functionality, which requires extensive training to utilize [61]. Other methods include knowledge engineering approaches such as WestLaw's Key Numbers system, which organizes legal resources (through manual means) into a knowledge graph based on legal topics and subtopics for structured information retrieval [67, 44]. However, in recent years there has been much exploration into incorporating machine learning empowered NLP techniques for semantic search in effort to improve upon deficiencies of lexical methods[36, 72, 8].

Generally, more advanced methods of information retrieval aim to encode tokens and spans into a vector space, where the notion of similarity between text is given by the distance between the documents in this vector space – a common method being cosine similarity [61]. While lexical methods such as TF-IDF or Bag-Of-Words aim to create these vectors using coarse statistical means, innovations in machine and deep learning allow for the development of more nuanced embeddings that capture deeper linguistic and semantic relationships. These methods go beyond mere word co-occurrence, learning representations that reflect the syntactic roles and semantic meanings of words within vast amounts of text [61, 59]. With the advent of transformer models, there has been much research into the usage of pre-trained transformer-based LLMs such as BERT for the generation of dynamic and context-aware vector embeddings.

2.5.2 Defining the Task of Information Retrieval

There are wide range of approaches for the task of information retrieval largely due to the ambiguous nature of the task. In particular, it is not trivial to define what exactly information retrieval training data would consist of, what defines a query or search result, and what defines the notion of relevance between a query and a resource. For instance, the Competition On Legal Information Extraction and Entailment (COLIEE) 2020 [31], defines two different tasks of legal information retrieval. The first task is case document retrieval where queries are a case document, and query results are other *relevant* cases. The second task is based on a Japanese Bar exam dataset where queries are questions taken from a bar exam regarding Japanese Civil Code, and query results are related Japanese Civil Code articles [31]. The competition defines two different notions of relevance between a query and resource for each of the two tasks. For the case document retrieval task, relevance is based on a notion of "notice", which is a formal legal term that denotes a legal case that is considered pertinent (i.e should be *noticed*) to a query case[31, 51]. For the second task, relevance is based on logical entailment, where the task is to determine whether or not some subset of Japanese Civil Code articles (AKA premises) entail a bar exam question (AKA hypothesis) or not.

Opijnen et al. (2017) [52] formalizes a number of other notions of relevance including the following:

1. Topical Relevance : where a query and resource independently pertain to the same subject matter or topic without necessitating the existence of a direct relationship between the query and resource such as with entailment.

2. Algorithmic Relevance : A quantitative notion of relevance based on a computational relationship between a query and information object defined by an algorithm. This could include the notion of frequency-based term importance as defined by TF-IDF, or more nuanced notions of semantic similarity between deep-learned vectors.

3. Cognitive Relevance : concerns the relation between a specific information need of a user and the information objects/resources. Unlike other measures such as entailment, topical relevance, or cognitive relevance, this notion is dependent on the user and their intent.

In particular, these various of notions can be viewed as different dimensions for an overarching concept of relevance, that when considered together, constitutes a more holistic and practical information retrieval system [52].

The task of information retrieval is multi-faceted and non-trivial. The various interpretations of the task and notions of relevance must be considered when designing an information retrieval system. At the end of the day, it consists of a family of sub-tasks that all aim to address some information need based on a notion of relevance.

2.5.3 Approaches

Approaches today for legal NLP tasks often investigate the usage of the BERT transformer architecture. Katz et al. (2023) [30] offers a keyword analysis into an exhaustive collection of Legal NLP publications, finding that keywords “BERT”, “attention” and “transformer” are among the most frequent terms to occur in publications.

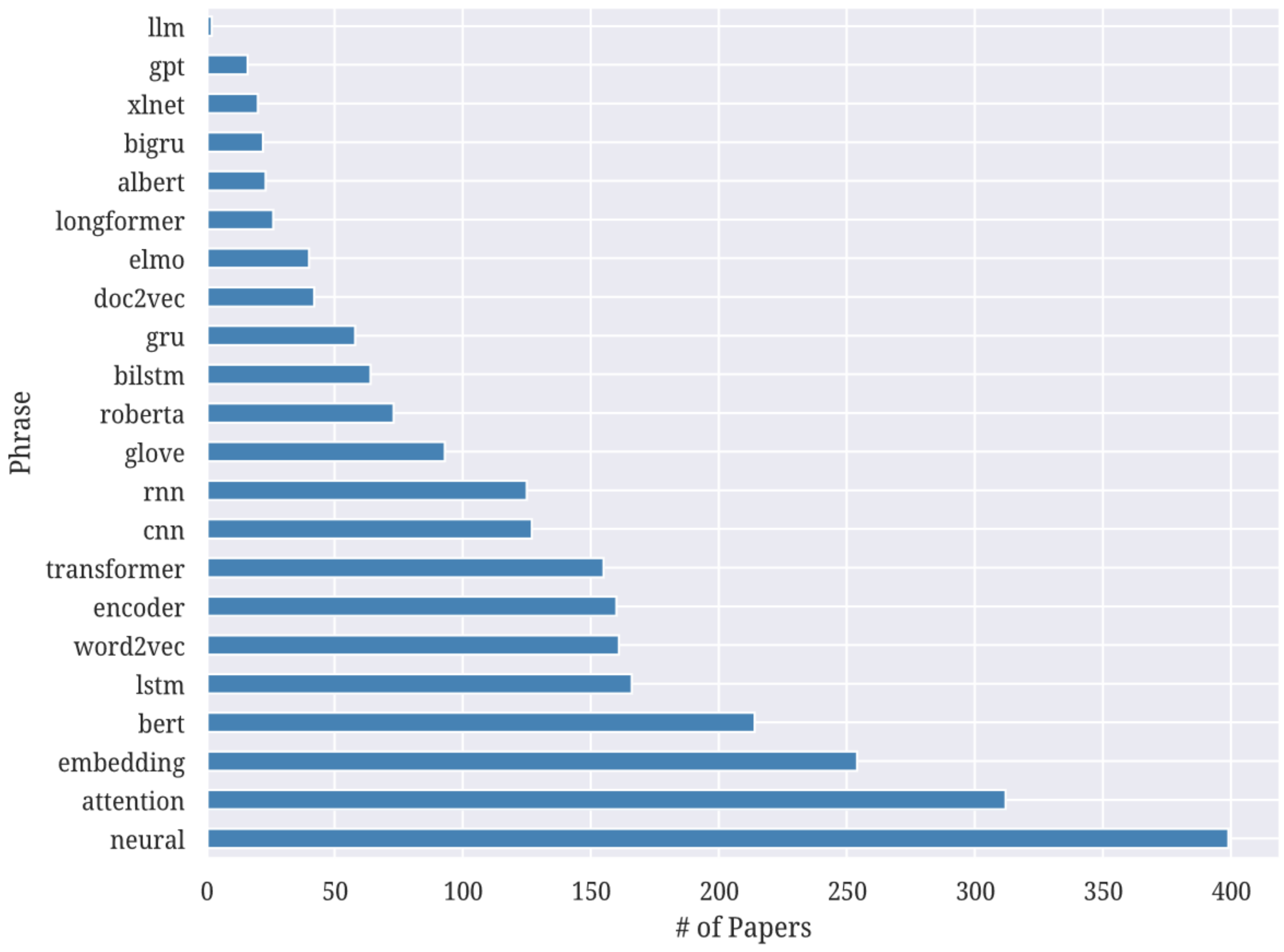


Figure 2.5: Distribution of Phrases in Legal NLP Publications [30]

Approaches leveraging the BERT model frequently feature a two-stage approach starting with a further pre-training phase (using a task such as MLM) to increase domain specificity, followed by task-specific training as evident in [49, 31, 50, 43]. This is often an important component when applying large language models for specialized domains and has been shown to be effective with domain-specific models such as SciBert [3] and BioBert [35], fine-tuned for the general science domain and biomedical domain respectively.

Information retrieval tasks are generally centered around understanding relationships between sequence pairs, regardless of the type of sub-task. When utilizing BERT, an important design decision is how to encode sequence pairs for training, in particular whether to use cross-encoding or bi-encoding (AKA dual-encoding).

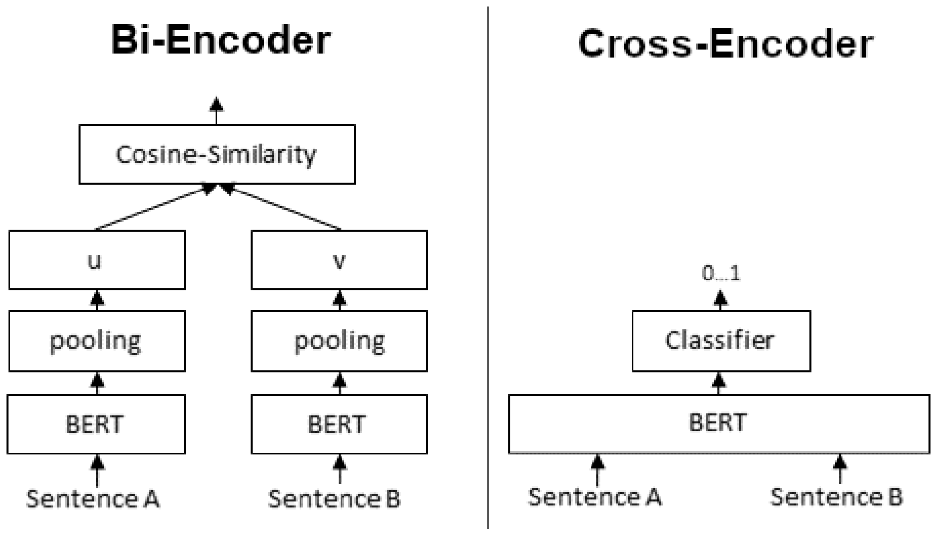


Figure 2.6: Cross-Encoder versus Bi-Encoder Architecture [12]

Using a bi-encoder architecture, the two sequences (i.e query and result) are encoded into embeddings independently and compared for similarity/relevancy [49, 12]. In this case, the attention mechanism of the BERT encoding layers attend to the words in each sequence separately – so tokens in one sequence have no direct interaction with tokens in another sequence, limiting contextual information. Because bi-encoding processes each sequence independently, the self-attention mechanism can be parallelized for each sequence, leading to faster computation times [49, 12]. Cross-encoding on the other hand combines the two sequences into a single sequence, allowing the self-attention mechanism to attend to tokens across both sequences. This provides the encoding of tokens with richer contextual information, allowing the model to capture more nuanced relationships between both sequences [49, 12]. However, this increases the complexity of the self-attention mechanism, which in turn adds computational complexity in comparing the two sequences compared to bi-encoding [49, 12, 58]. Moreover, in the context of an information retrieval system, combined query and document encodings must be generated dynamically for every query as opposed to a bi-encoding scheme where documents can be pre-encoded independently at one time, stored, and re-used for queries. The choice of encoding method depends on balancing richer, nuanced encoding versus faster computation times. Bi-encoders are preferred for dense document or sequence retrieval tasks, facilitating efficient comparison in large search spaces. Conversely, in complex language domains with rich inter-dependencies, such as legal documents, cross-encoders offer increased nuance and are beneficial.

This choice between the various encoding approaches complicates the notion of fine-tuning a model. As mentioned above in section 2.3.1, when fine-tuning a model for a specific task, there are two components of a model that undergo weight-adjustment. The first are the core Transformer encoding layers which are responsible for producing rich context-aware embeddings for tokens and sequences. The second component is the task specific head itself (e.g. a classification layer). Consequently, for the task of information retrieval, if the goal is to utilise a bi-encoding scheme, the goal of fine-tuning is to improve the standalone representational capacity of the encoding layers. As such, the choice of fine-tuning tasks and training data guides the encoder training to produce more useful encodings for the fine-tuning task. Within an information retrieval system, resources (e.g documents) are encoded independently and placed into an index. When search is performed, the query is then encoded independently and compared against the each resource encoding in the index to determine which resources produce the highest similarity (e.g. cosine similarity).

If the goal is to utilise a cross-encoding scheme, the goal of fine-tuning is both the improve the core encoder itself, but additionally to improve the performance of the task-specific head which is used directly for the ultimate information retrieval system. With this scheme, when a search is performed, the query is dynamically combined and encoded with each resource producing a combined embedding. Then the fine-tuned classification head is used to determine whether the two sequences (query and resource) are similar through its own notion of similarity as defined by its fine-tuning weight-adjustment process.

An interesting compromise that achieved impressive results in the COLIEE [31] and ALQAC [63] legal data processing competitions was to combine lexical methods such as TF-IDF with a LLM-based cross encoding scheme. In particular, a common pipeline was to use a lexical method such as TF-IDF to determine an initial top K relevant results with respect to a query, then with this smaller search-space of results, utilise a fine-tuned cross-encoder to either re-rank the K results or further filter the results down to a smaller subset [63, 31, 50, 49]. This method serves to decrease the computational complexity through using a lightweight lexical method, while also taking advantage of the nuanced capabilities of cross-encoding. Other strategies focus on combining bi-encoders with cross-encoders, offering another perspective in balancing computational cost with more nuanced analysis [53].

Luo et al. (2023) [43] offers an interesting solution to address the issue of the non-trivial nature of information retrieval and relevance mentioned above in section 2.5.2. The paper introduces a multi-task fine-tuning approach, to holistically increase domain knowledge in an effort improve semantic representation ability as well as understanding of user intent behind queries [43]. In particular, the experiment utilises varying combinations of the tasks of masked language modelling (MLM), next sentence prediction (NSP), semantic textual similarity (STS), and textual entailment (TE) to fine-tune the core encoder layers of a BERT model for the task of retrieval-based question and answering. In this, case the multi-task scheme is used as a form of further-pretraining, whereby using a variety of tasks-specific fine-tuning, a the study is able to improve a model's generation of standalone embeddings for the task of retrieving answer in a corpus that similar to a given question. The study found that the combination of fine-tuning BERT with MLM and STS produced the best results, outperforming the base BERT model and single-task fine-tuned variations [43].

Chapter 3

Fine-Tuning the BERT Model

The aim of the fine-tuning stage of this project is to explore what approaches are optimal for improving the two components of a model that undergo weight adjustment through fine-tuning : the underlying transformer encoders, and the task specific heads. As noted above in Section 2.5.3, a critical decision in designing an information retrieval system is the choice between a cross-encoder or dual-encoder architecture. There are many implementations using either architecture, or combining these architectures with each other or with other lexical methods. By investigating what fine-tuning approaches are optimal for both the core encoding mechanism and task-specific performance, we establish foundational design decisions that are pertinent to both bi-encoder and cross-encoder architectures, regardless of implementation. The contributions of this fine-tuning exploration can be visualized in Figure 3.1 below.

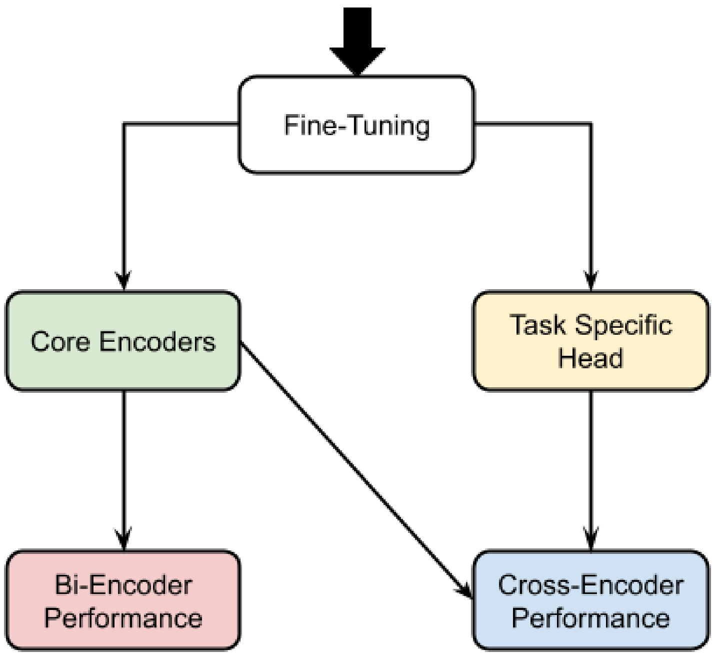


Figure 3.1: Fine-tuning contribution graph. Arrows indicate how one component contributes to performance of another component(s)

There are essentially two challenges that this project aims to address through fine-tuning the pre-trained BERT model. The first is to increase the domain specificity of the core Transformer model for the legal domain. This specifically targets the underlying encoder layers and attention mechanism of the Transformer architecture to better capture the complexities of legal language to improve the generation of representations. The second is to increase the task-specific performance of the model for information retrieval. An essential component to this challenge is to address the non-trivial and multi-dimensional nature of the task of information retrieval and relevance as mentioned in section 2.5.2). This includes methods to improve the performance of a task-specific head for the task of information retrieval, as well as to use the task-specific fine-tuning to guide the adjustment of the core encoding layers to produce more effective representations for the task.

As noted in Section 2.5.3, there are a number of fine-tuning approaches for increasing domain specificity of a pre-trained BERT model. Generally using in-domain training data will have a beneficial effect, however, certain approaches align particularly well with the Transformer self-attention mechanism, namely, masked language modelling. Accordingly, an essential component is to investigate the benefit of utilising MLM for increasing domain-specificity compared to directly fine-tuning the pre-trained BERT model with task-specific methods for information retrieval. In doing so, we paint a picture of the utility in introducing *further-pretraining* as defined by Gururangan et al. (2020) [24] using the MLM task, before fine-tuning the model for the specific downstream task (ie. information retrieval).

Next to determine how to approach fine-tuning for the specific task of information retrieval. As such, this necessitates consideration of what is sufficient to capture the multi-faceted nature of information retrieval and the notion of relevance. Most approaches are one-dimensional, focusing on a specific interpretation of the task whether it be semantic entailment between documents, generative or extractive question and answering, or semantic textual similarity. As such, it is necessary to investigate the benefit that fine-tuning on multiple tasks provides in comparison to single-task approach, and hence determine if a single-task is sufficient. Additionally, this warrants comparing sequential and parallel multi-task fine-tuning to investigate the effect of negative transfer and catastrophic forgetting.

So this establishes a number of investigation points to paint a picture of what is most effective to fine-tune for the task of information retrieval :

1. Evaluate the effect of a number of fine-tuning approaches for improving the core encoding function of the BERT model for the task of information retrieval as to explore what is most effective for a bi-encoding approach.

2. Evaluate the task-specific performance (combination of both core encoding ability and task specific head performance) of a number fine-tuning approaches to explore what is most effective for a cross-encoding approach.

3. Evaluate the effect of introducing masked language modelling further pre-training to increase domain specificity.

As such, the various fine-tuning approaches can be distilled into a number of models that have been implemented and evaluated:

Model	Description
`Single_TE`	Fine-tuned for single task of Textual Entailment (TE)
`Single_QA`	Fine-tuned for single task of selective question and answering (QA)
`Multi_Parallel`	Fine-tuned in parallel using tasks of TE and QA
`Multi_Sequential`	Fine-tuned using tasks of TE and QA sequentially

Table 3.1: Caption

This chapter is organized into the design and implementation of each type of model, followed by an evaluation of how each model performs in regards to core encoder performance and task specific head performance for the task of information retrieval.

3.1 Work Flow

To compare and contrast a number of fine-tuning strategies, I created fine-tuning routines for each strategy to streamline the process of creating a number of fine-tuned models that can be compared and evaluated. Each routine follows the same general structure and order of operations. The first stage is to prepare the training data for the fine-tuning routine. This includes any data cleaning or re-formatting that may be necessary for the given training task, as well as tokenizing the data and establishing the training, validation, and test dataset splits. This stage also involves initializing a data loader for the datasets, enabling the dynamic loading of input samples and labels for training, validation, and testing routines, a practice commonly adopted for memory efficiency. The next stage is to initialize a training loss function, optimizer, and learning rate scheduler, followed by the actual training routine. Each fine-tuning routine follows a standard structure where training is executed for some number of epochs, where within each epoch we execute both a training loop for weight optimization and a validation loop for hyper-parameter tuning and early stop detection. Finally, at the end of each fine-tuning routine, we evaluate the tuned model against the test dataset.

In terms of work flow, Jupyter notebooks are used to implement the fine-tuning routines and evaluation of the models. Using the `Transformers` library, every BERT model is implemented as an object which inherits from the `PreTrainedModel` [46] class of the `Transformers` library. This setup is particularly advantageous because it simplifies the saving of each model's configuration and weights into specific directories after fine-tuning. This allows for a seemless, plug-and-play scheme where fine-tuned models can be easily loaded back into programs for making predictions, running evaluation tests, or for further fine-tuning. This method is highly flexible and well-suited for the exploratory nature of our project, enabling the sequential fine-tuning of a base model for various tasks. Essentially, a model can be fine-tuned for one specific task, saved, and then reloaded to undergo fine-tuning for a different task, thereby generating a series of progressively refined models. All experiments were optimized for and run on a GPU lab machine provided by the School of Computer Science in order to leverage the GPU-based hardware acceleration capabilities offered by PyTorch. In particular the machine utilises an Nvidia GeForce RTX 3060 12GB GPU with 16GB of RAM.

3.2 Masked Language Modelling

The goal of the masked language modelling procedure is to increase legal domain specificity. The legal domain of course is rather heterogeneous, consisting of a large number of documents types with varying levels of divergence from the colloquial domain as discussed in section 2.1. As a result, to assemble a comprehensive training dataset, sentences were extracted from a diverse selection of English legal datasets, namely :

1. UK Legislative Documents [6]
2. Australian Court Case Summaries [20]
3. US Court Case Holdings [76]
4. US Congressional Bill Summaries [33]
5. r/legal_advice Reddit Community posts [37]

Once extracted, the sentences were randomly shuffled before creating a final dataset for fine-tuning and evaluation. The BERT model has a maximum input size of 512 tokens. As such, when constructing a dataset for fine-tuning BERT, it is essential to consider the lengths of inputs. When it is the case that a training dataset has inputs that exceed this length, a truncation strategy must be applied. This however was not an issue for the 5 datasets listed above as can be seen in Figure **??** depicting sentence length distribution.

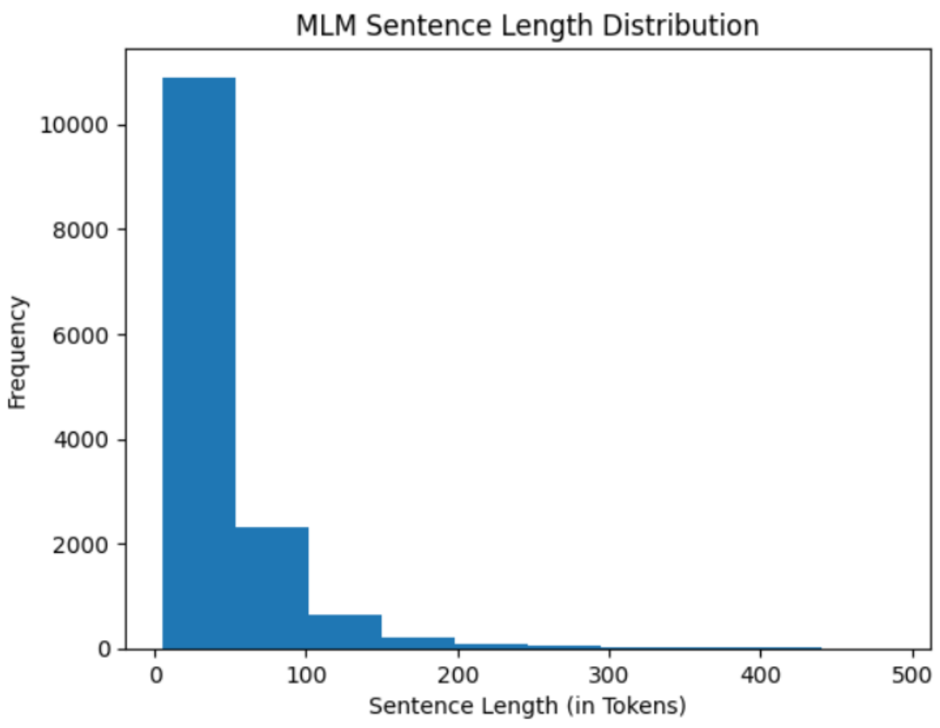


Figure 3.2: Sentence length distribution of MLM dataset (in tokens)

To mask tokens in each sentence, a `DataCollator` [15] is used to randomly mask 15% of tokens in each sentence, which is the same configuration used to pre-train the BERT model [17]. With this, random tokens in each sentence are replaced with a special `[MASK]` token which is unique to BERT for the MLM task. When used in conjunction with a `DataLoader` object, samples are masked dynamically as they are loaded during training, validation, and testing routines. In total, the routine makes use of 15 thousand samples with a 70/15/15 split. I've established via trial and error that a mini-batch size of 8 samples is the maximum that reliably prevents memory issues during training (for my specific hardware).

The training loop utilizes the cross-entropy loss function, AdamW optimizer, and a linear learning rate scheduler with warm-up. As the MLM task is a multi-class classification task, the natural choice for loss function was cross-entropy [13]. AdamW is common and effective choice for mini-batched stochastic gradient descent [32, 41]. This optimizing/gradient-solving method combines the benefits of adaptive learning rates with a more robust approach to weight decay regularization, leading to improved model generalization and faster convergence – particularly important for deep models such as BERT [32, 41]. This strategy is especially beneficial for complex language models, where navigating large, sparse datasets and preventing over-fitting are critical for achieving state-of-the-art performance [41]. A linear learning rate scheduler with warm up is used with an initial learning rate of 3e-5. This seems to be common practice for LLM's as noted by Jin et al. [28] and is utilised by both BERT [17] and GPT-3 [4]. With these configurations, the routine runs for 5 epochs and utilises an early stopping mechanism based on validation accuracy. In particular, the routine stops if it detects 3 epochs of constant decrease in validation accuracy. This was implemented along with a model checkpoint system to save the state of the model with the highest validation accuracy across all epochs.

3.2.1 MLM Fine-Tuning Evaluation

To assess the performance of the fine-tuned MLM model, I used top-k accuracy and perplexity and bench marked the model against a number of alternatives. When given a sentence with masked tokens, for each masked token, the model creates a probability distribution over the entire vocabulary representing the probability of each type being the correct token that was masked. The top-K metric refers to the rate at which the correct term is predicted to be within the top K most probable terms in the prediction distribution. Top-K accuracy is preferred over standard accuracy (i.e top one accuracy) because it is more informative for this scenario. The task of MLM is rather difficult due to the complexity and ambiguity of natural language. As such, normal accuracy may unfairly penalize a model for choosing synonyms or phrases that are contextually appropriate but not the exact match expected. Top-k accuracy reflects an MLM's ability to understand and predict multiple valid alternatives rather than penalizing it for not choosing the single correct answer.

Perplexity is a natural choice for evaluating Masked Language Models (MLMs) as it quantifies model uncertainty in generating text. It reflects the model's ability to understand and predict language by calculating the average negative (log) likelihood of a sequence of words, with lower perplexity indicating better predictive performance.

$$perplexity(w_i) = \exp\left(-\frac{1}{N}\sum_i log(P(w_i))\right) \tag{3.1}$$

Figure 3.3: Perplexity Equation

Since MLMs aim to predict masked words within a context, perplexity directly assesses their effectiveness in capturing linguistic patterns and coherence. Thus, it provides a clear, quantifiable measure of how well an MLM grasps the complexities of a language. A higher level interpretation is the metric measures how "perplexed" the model is seeing a token within a language.

I benchmarked the fine-tuned MLM model against three other models to elucidate the effect of increasing domain specificity through MLM fine-tuning :

1. `bert-base-uncased` : the base pre-trained BERT model
2. `legalbert` [5] : a top of line BERT model fine-tuned for the legal domain. This model is considered highly effective in regards to domain specificity and is one of the most cited models in the legal NLP space [30]
3. Witten-Bell Trigram : a more traditional and basic approach for language modelling utilising a frequentist approach to model word transitions using Witten-Bell smoothing

To evaluate the other BERT models (`legalbert` and `bert-base-uncased`) fairly, an outer MLM classification layer is trained using the MLM dataset, with the base encoder weights of the models frozen. This allows the evaluation to isolate the effect of the core encoder weights in performing the task of MLM. So each model has the same fine-tuned MLM classification head, but with different core encoder weights. This paints a better picture of how the MLM task adjusts the core encoder weights to increase domain specificity. The Witten-Bell Trigram model is trained on the same training dataset split.

Model	**Top 5 Accuracy**	**Top 10 Accuracy**
MLM Model (Best)	84.77	88.62
`bert-base-uncased`	74.32	78.26
`legalbert`	82.20	86.20
Witten-Bell Trigram	53.27	55.56

Table 3.2: Top-5 and Top-10 Accuracy Results (each value is average of 5 runs)

Model	**Perplexity**
MLM Model (Best)	4.45
`bert-base-uncased`	13.02
`legalbert`	6.73

Table 3.3: Perplexity results (each value is average of 5 runs)

The tables in Table3.2 and Table 3.3 show that the fine-tuned model performs the best both in terms of top-K accuracy and perplexity. This strongly suggests that the MLM task has a positive effect in increasing the domain-specificity of the model.

3.3 Single Task Classification

3.3.1 Textual Entailment Model

Datasets

For the textual entailment task, a dataset of United States case-holdings [76] is used. In this dataset each sample contains a court case citing-context, a corresponding case-holding along with a binary label indicating whether the two are relevant. A citing context in this case is text in a court document that precedes a court case citation and includes direct facts and contextual information describing a particular court case. A holding follows a citing context and reflects the legal conclusion relevant to the case or issue described in the citing text [76]. So the citing context can be viewed as premises or facts where the corresponding case holding is a logical conclusion from those facts. Hence this dataset is used for the task of textual entailment.

Context
They also rely on Oswego Laborers' Local 214 Pension Fund v. Marine Midland Bank, 85 N.Y.2d 20, 623 N.Y.S.2d 529, 647 N.E.2d 741 (1996), which held that a plaintiff "must demonstrate that the acts or practices have a broader impact on consumers at large." Defs.' Mem. at 14 (quoting Oswego Laborers', 623 N.Y.S.2d 529, 647 N.E.2d at 744). As explained above, however, Plaintiffs have adequately alleged that Defendants' unauthorized use of the DEL MONICO's name in connection with non-Ocinomled restaurants and products caused consumer harm or injury to the public, and that they had a broad impact on consumers at large inasmuch as such use was likely to cause consumer confusion. See, e.g., CommScope, Inc. of N.C. v. CommScope (U.S.A) Int'l Grp. Co., 809 F. Supp.2d 33, 38 (N.D.N.Y 2011) (<HOLDING>); New York City Triathlon, LLC v. NYC
Holding
holding that plaintiff stated a 349 claim where plaintiff alleged facts plausibly suggesting that defendant intentionally registered its corporate name to be confusingly similar to plaintiffs CommScope trademark
Binary Label
1

Table 3.4: Example Case Holding Sample

Model Configuration & Routine

The model used for this task consists of the 12 core BERT encoding layers with a sequence classification task-specific head which consists of two linear layers and a ReLu activation function. The configurations of training routine are as follows

Number of Samples	15000
Batch Size	16
Learning Rate	1e-5
Number of Epochs	5 with early stopping mechanism with patience of 3 epochs
Loss Function	BCEWithLogitsLoss (Binary Cross-Entropy with Logits)
Optimizer	AdamW
Learning Rate Scheduling	Linear with warm up
Data Split	70/15/15

Table 3.5: Textual Entailment Fine-Tuning Configuration and Hyper-parameters

3.3.2 Selective Question and Answering Model

Datasets

The goal of selective question and answering is to identify *evidence* within a corpus of information given a question. The dataset consists of user questions regarding the privacy policy of a number of technology companies [57]. Each sample consists of a user question, along with a segment of text from a company's privacy policy that either answers or does not answer the question. This is framed as an answer selection task since the goal is to identify a set of evidence (segments in a policy document) out of a pool of candidate pieces of evidence that support the question. Privacy policies are crafted to be legally precise in language, as they need to comply with laws and regulations. They are also intended to not only be read by legal experts, but also by laymen. The intuition in including this dataset and task is to introduce a model to diverse sentence constructions (questions written by laymen, and policy written by experts) likely to be met when using an information retrieval system.

Query
How long will my browser history be kept?
Segment
If you cease usage of the Cake browser, we will retain certain data for analytical purposes and recordkeeping integrity, as well as to prevent fraud, enforce our Terms of Service & License Agreement, take actions we deem necessary to protect the integrity of our service, or take other actions otherwise permitted by law.
Label
Relevant

Table 3.6: Example Selective Q&A Sample

Model Configuration & Routine

The model used for this task shares the same configurations as that for textual entailment in Figure 3.5.

3.4 Multi-Task Fine-Tuning

3.4.1 Sequential Multi-Task Fine-Tuning

The sequential fine-tuning structure is relatively straightforward. The variant of sequential fine-tuning explored switches between tasks after the completion of each task, as opposed to switching at a higher frequency such as per epoch. In particular, the sequential model was first trained using the textual entailment task, followed by the selective question & answering task.

3.4.2 Parallel Multi-Task Fine-Tuning

Data Preparation

To support parallel per-batch multi-task training on a per-batch basis, a custom `CombinedDataset` class is designed to manage the two distinct datasets for Textual Entailment (TE) and Question and Answering (Q&A). This class tokenizes the data from the separate DataFrames for TE and Q&A, based on a given data split (training, validation, or testing). A custom getter function fetches processed samples from both datasets, providing tokenized input IDs, attention masks, token type IDs, and labels for both datasets, thus enabling simultaneous training on TE and Q&A within a single model framework. Due to the fact that batches from two datasets are loaded per training iteration, a reduced batch size of 8 samples is used rather than the 16 samples used for the other fine-tuning configurations due to hardware constraints.

Model Configuration & Fine-Tuning

To configure a model capable to training on each task in parallel at each batch, a custom model was implemented. In particular the model inherits from the `nn.Pretrained` class of the transformers library where custom task-specific heads were defined for each task and a `forward` prediction function. For each batch, a prediction for each task is made using each of the two task-specific heads. Then binary cross entropy loss is computed between the models raw logit prediction and its labels. The loss for both of the tasks are averaged, where the gradient of this combined average loss is used for weight optimization, a generally effective approach as explored by Don-Yehiya et al. (2023)[18].

3.5 Evaluation

The performance of the evaluation of the three models are three fold. First we evaluate the difference in task-specific performance of each model, evaluating the performance of each fine-tuned task-specific head (or heads in the case the multi-task models) in determining whether two sequences are related or not. Next, we evaluate the differences in performance between the core encoding layers between each of the models. Finally, we evaluate the differences in performance after using MLM to further pre-train the highest performing model of the two first evaluation stages to investigate the effect of increasing domain specificity from the MLM task. The evaluation performed are not to provide ground-truth measures of the performance of each model, but rather to show the relative differences between the models in a controlled setting.

3.5.1 Evaluating Fine-Tuned Task-Specific Heads

To evaluate and compare each of the models, I ran each model against the test set of each of two tasks (TE & selective Q&A) and computed the average accuracy across each dataset. Evaluation metrics for the `bert-base-uncased` model are also included as a benchmark. In this case, an untrained task-specific head is attached to the model for the two classification tasks.

Model	TE Accuracy	Q&A Accuracy	Averaged Accuracy
`Single_TE`	**82.93**	76.84	79.89
`Single_QA`	79.96	**82.53**	81.245
`Multi_Parallel`	80.44	79.42	79.93
`Multi_Sequential`	71.64	81.38	76.51
`bert-base-uncased`	73.73	81.78	77.755

Table 3.7: Accuracy Results of Fine-Tuned Task-Specific Heads for TE and selective Q&A

Table 3.7 shows the accuracy of each model against each of the two fine-tuning tasks. First examining the accuracy for the TE task, unsurprisingly the `Single_TE` model trained solely on the TE task performs the best. The `Multi_Sequential` model performs the worst by a significant margin for the TE task. At first glance, this result would be indicative of the catastrophic forgetting phenomenon, whereby first fine-tuning the sequential multi-task model on TE, then subsequently fine-tuning using the selective Q&A task, the model loses the information learned from the TE task. However, interestingly the `Single_QA` model performs significantly better. A possible explanation is that by first fine-tuning the model on TE, a bias is introduced to the model's parameter space causing the model to start in a biased location in parameter space. The `Single_QA`, lacking this initial bias, might explore a different part of the parameter space that, while optimized for the selective Q&A task, doesn't preclude a decent performance on TE due to different initial conditions and learning dynamics [39, 70]. Whether or not this effect takes place in the parameter space of the classifier head or within the deep encoder layers requires further exploration.

Looking at the results for the selective Q&A task, the `Multi_Sequential` model performs rather well with the second highest accuracy which is somewhat counter-intuitive. A possible explanation is that the corruption of parameters by introducing the sequential learning of the two tasks hurts the models generalizability, resulting in worse performance for the TE task, but still over-fits to the selective Q&A task due to the fact that the Q&A fine-tuning is used more recently. Unsurprisingly, the `Single_QA` model performs the best for the selective Q&A task.

Overall for average accuracy, the `Single_QA` model performs the best with the `Multi_Parallel` model coming in second with overall consistent metrics for each task. The performance metrics of the `Multi_Parallel` are indicative of the aforementioned regularization effect, whereby utilizing multiple fine-tuning tasks, we alleviate over-fitting to a specific task while improving domain specificity. The `Multi_Sequential` model overall exhibits the worst performance, with lower accuracies than the `bert-base-uncased` model, which is indicative of a combination of catastrophic forgetting and poor handling of contrasting representations and loss gradients between the two tasks. It is worth noting that the results of the `Multi_Sequential` are largely dependent on the order at which the fine-tuning tasks are introduced.

3.5.2 Evaluating Fine-Tuned Core Encoder

To evaluate the performance of the core-encoder layers of each fine-tuned model, a strategy was employed to separate their impact from the fine-tuned task-specific heads. To match the multifaceted nature of information retrieval, a combined dataset encompassing various sub-tasks is used. This dataset consists of unseen TE data of case-holdings, unseen selective Q&A privacy policy data, and additionally a new legal contract dataset, LEDGAR [65], consisting of samples containing legal contract text along with a number of topical keywords describing the contract.

The evaluation process for each model is to attach a task specific head for sequence classification to each model's core encoding layers. The newly attached classification head is then fine-tuned using a 70% training split with the following configurations/hyper-parameters :

Number of Samples	15000
Batch Size	16
Learning Rate	1e-5
Number of Epochs	5 with early stopping mechanism with patience of 3 epochs
Loss Function	BCEWithLogitsLoss (Binary Cross-Entropy with Logits)
Optimizer	AdamW
Learning Rate Scheduling	Linear with warm up (5% of training data)
Data Split	70/15/15

Table 3.8: Fine-Tuning Configuration and Hyper-parameters for Evaluation of Model Core Encoding Layers

To isolate the contribution of the core encoding layers of each model for classification performance, their weights are frozen during the fine-tuning process. This prevents adjustments to the weights, ensuring that only the classifier layer is tuned. This approach allows for a clear assessment of how each model's encoding layers affect classification performance. It's important to note that for each model, the peak validation accuracy was reached in the third epoch. Therefore, the state of each models at the third epoch is used for evaluation across all models. Finally each model is evaluated against the test split of the combined dataset.

Model	**Accuracy**	**Precision**	**Recall**
`Single_TE`	87.68	85.73	79.34
`Single_QA`	75.78	79.23	45.23
`Multi_Parallel`	90.76	90.57	75.67
`Multi_Sequential`	83.96	82.78	70.54
`bert-base-uncased`	83.42	80.61	71.68

Table 3.9: Evaluation metrics of each model's fine-tuned core encoding layers. Depicted are accuracy, precision and recall metrics of each model after freezing core encoding layers and training a task-specific head (classifier) on a combined information retrieval dataset

Table 3.9 shows the metrics from evaluating the fine-tuned encoder layers of each of the models. The Multi_Parallel exhibits significantly better metrics than than the other models. This would suggest that fine-tuning with a combination of multiple tasks for fine-tuning produces a regularization effect, allowing the core encoding layers of the model to learn more robust and general representations of in domain text, preventing over-fitting to a particular task of dataset. Hence the combination of the robust embeddings along with a fine-tuned task-specific head produces the highest metrics.

Interestingly the Single_TE also performs quite well, coming in second for accuracy, surpassing the Single_QA and Multi_Sequential models. This is likely due to the case holding corpus (used for the TE task) more closely aligning to the combined evaluation dataset compared to the selective Q&A dataset. The selective Q&A dataset, which the Multi_Sequential and Single_QA models are fine-tuned on, is less technical in legal language (as mentioned in section 3.3.2), consisting of text that is legally precise, but designed to be understandable for laymen. The introduction of the LEDGAR [65] dataset in the combined dataset aligns more closely with the case-hold data used for the TE task, as both datasets consist of highly technical language designed for specialists within the domain.

3.5.3 Effect of Introducing MLM

An additional model was created by introducing an initial MLM fine-tuning routine to the Multi_Parallel model. In particular, the fine-tuned MLM model (from Section 3.2) is further fine-tuned using the parallel multi-task procedure. The resulting model, MLM + Multi_Parallel, was then evaluated using the same two procedures to investigate the difference in performance of the task-specific head and underlying core encoding layers.

Model	TE Accuracy	Q&A Accuracy	Averaged Accuracy
MLM + Multi_Parallel	**86.01**	**81.1**	**83.55**
Multi_Parallel	80.44	79.42	79.93

Table 3.10: Accuracy Results of parallel multi-task model with MLM pre-training (MLM + Multi_Parallel) compared to results without MLM pre-training (Multi_Parallel)

Table 3.10 shows how introducing additional pre-training improves performance of the parallel multi-task model across both tasks, indicating a positive effect of utilising MLM pre-training to increase domain-specificity when used in-conjunction with a fine-tuned task-specific head.

Model	Accuracy	Precision	Recall
MLM + Multi_Parallel	88.98	90.31	70.17
Multi_Parallel	**90.76**	**90.57**	**75.67**

Table 3.11: Evaluation metrics for relative performance of core encoding layers of parallel multi-task model with MLM pre-training (MLM + Multi_Parallel) and without (Multi_Parallel)

As shown in Table 3.11, the MLM + Multi_Parallel exhibits a slight decrease in core encoder performance. This is worth noting that this is could easily be affected by dataset selection and size for the MLM task and for the combined evaluation dataset.

Chapter 4

Application

A user application is included to implement a usable search engine over a corpus of legal documents utilising the highest performing model from the fine-tuning stage of this project. This component of the project involves the creation of a user interface, a backend which implements a document search pipeline, and a REST API to facilitate communication between the interface and the backend.

A key challenge in using deep learning models for semantic search involves finding a balance between efficiency and the complexity of using such sophisticated models to search through a large dataset and search space. As discussed in section 2.5.3, an important consideration when using deep learned encoding models for information retrieval is the choice between using cross-encoding and bi-encoding. While cross-encoding allows for more nuanced analysis of the interactions between a query and result, it is much more computationally expensive compared to bi-encoding. As a result, two different search pipelines have been implemented to tackle this issue.

I have selected and compiled a dataset of 5 thousand court case documents to serve as a corpus to search over for proof of concept.

4.1 Pipeline Approaches

I implemented two search pipelines for the application, one utilizing a combination of TF-IDF similarity scoring and cross-encoding using the pre-trained BERT model, and the other a combination of TF-IDF and dual-encoding. For each of the two pipelines, TF-IDF is used to retrieve an initial subset of relevant documents. As a computationally cheaper method for similarity scoring between a query and corpus of documents, this helps reduce the computational cost of utilizing a deep fine-tuned transformer model to perform nuanced vectorization over a large search space of documents. After retrieving a top-K number of most relevant documents, the resulting documents are passed to the fine-tuned BERT model for more nuanced encoding and analysis. Then the documents are re-ranked using the fine-tuned BERT model, where this final subset of ranked documents are returned to the user as results.

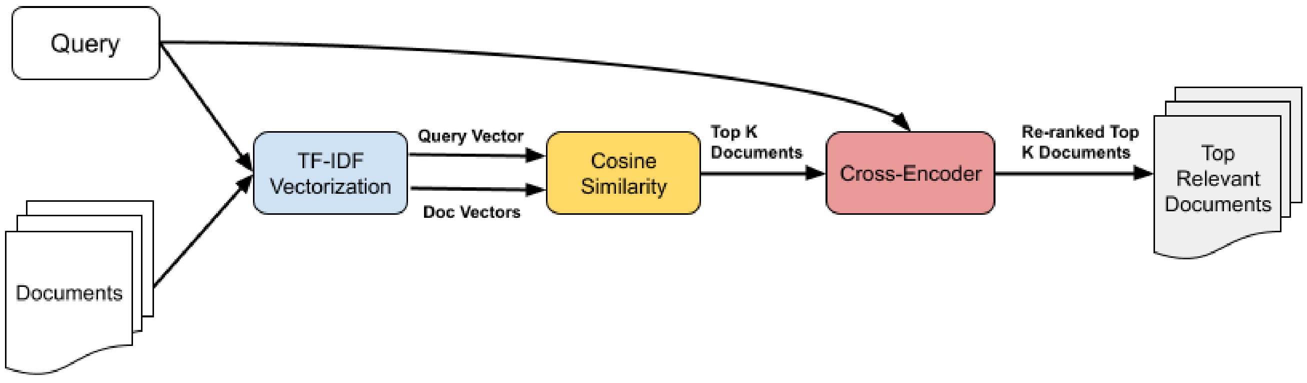


Figure 4.1: Cross Encoder Search Pipeline

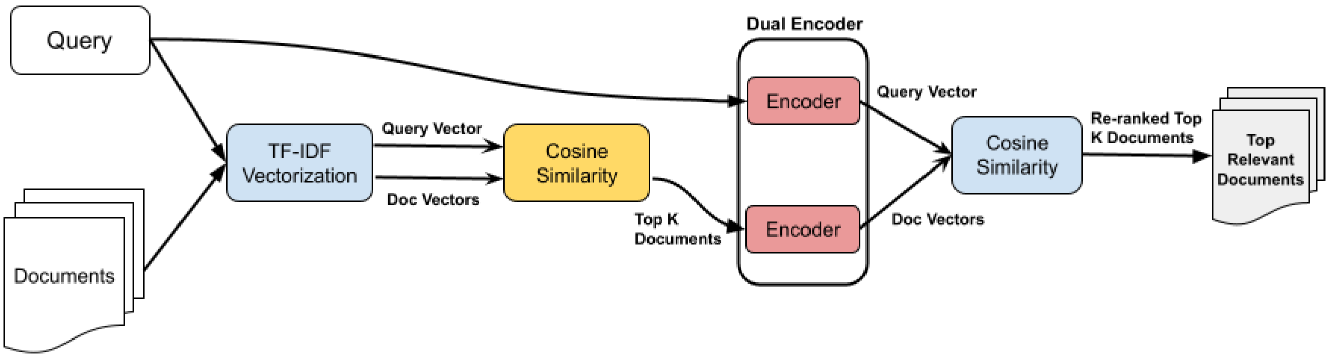


Figure 4.2: Dual Encoder Search Pipeline

4.1.1 TF-IDF Engine

The TF-IDF functionality is self-contained and implemented as a class called `tfidf_corp`. When initialized, the object ingests a corpus of documents. The ingestion process can be viewed as a pipeline and begins with a document truncation and division process. In particular, the long and dense nature of legal documents complicates the generation of vector representations that can capture the syntactical or semantic meaning of the document. In many cases, a query may only be relevant for a small subsection of a long legal document, where any extra text outside of the relevant subsection are effectively noise. To mitigate this issue, the TF-IDF engine implements a strategy that breaks each document that has a length over a specific threshold, down into separate subsections. In particular, each document longer than 1024 characters is divided into subsections according to a particular stride length. This means that each document longer than 1024 characters is split into subsections that overlap by 128 characters. Extra care is taken to ensure that the document is not split in the middle of a sentence, which could complicate the natural understanding of the pre-trained model. Breaking down each document also addresses the input size limit of BERT which is 512 tokens.

After applying the truncation and division strategy the next stage is to pre-process the document text for vectorization. The pre-processing steps implemented are tokenization, removal of stop words, lemmatization, and stemming. Finally, the documents are vectorized into a TF-IDF matrix. In order to keep track of which subsections belong to which overarching document, a separate data frame is maintained, mapping each subsection to a corresponding document identifier. The resulting TF-IDF matrix is then saved to a `pickle` file for persistent storage and loading to circumvent re-generating the matrix each time the search engine is initialized.

When a query is searched, the same pre-processing pipeline is applied to tokenize and clean the query text. Next the query is converted into a TF-IDF vector using the same vocabulary and IDF values computed from the corpus. This step is crucial to ensure that the query and the documents in the corpus are represented in the same feature space.

Using the mix of lexical and semantic based approaches presents two benefits. The first is by initially using a lightweight lexical method to retrieve an initial subset of results, the search space for the deep-learned cross-encoder is significantly reduced to the lexically most relevant results, improving computation time [49, 50]. The second benefit is that lexical analysis, while more coarse than deep-learned methods, does offer a useful dimension of relevance through keyword analysis. Viewing the task of information retrieval from a practical point of view, exact key word matching is often an essential need for lawyers in addition to capturing deeper semantic components such as intent. This mixture can be viewed as an alternative balance between precision and recall of the information retrieval system. While deep-learned semantic approaches are able to capture broader semantic connections that lexical methods fall short on, they can alternatively identify too many connections that a user of the system may not be interested.

4.1.2 BERT Cross Encoder Engine

After retrieving an initial top K results from the TF-IDF engine, the cross encoder then performs semantic analysis between the query and each of the top K lexically most relevant document results. Each result is concatenated with the query, producing a combined sequence. The combined sequence is then tokenized and passed to the cross-encoder model where the sequence undergoes a series of transformations to ultimately produce a cross-encoding that captures the syntactical and semantic interactions between the query and document result. This encoding is then passed to the task-specific head (sequence classifier) which produces raw logit predictions for each class of the classification head (positive/relevant or negative/irrelevant). A softmax operation is applied over the logits to convert them to probability distribution over the two classes, where the probability of the positive class is extracted and used as a similarity/relevance score. The top K documents are then re-ranked using these computed similarity scores to produce the final ordering of results, ranked by semantic relevance.

4.1.3 BERT Bi-Encoder Engine

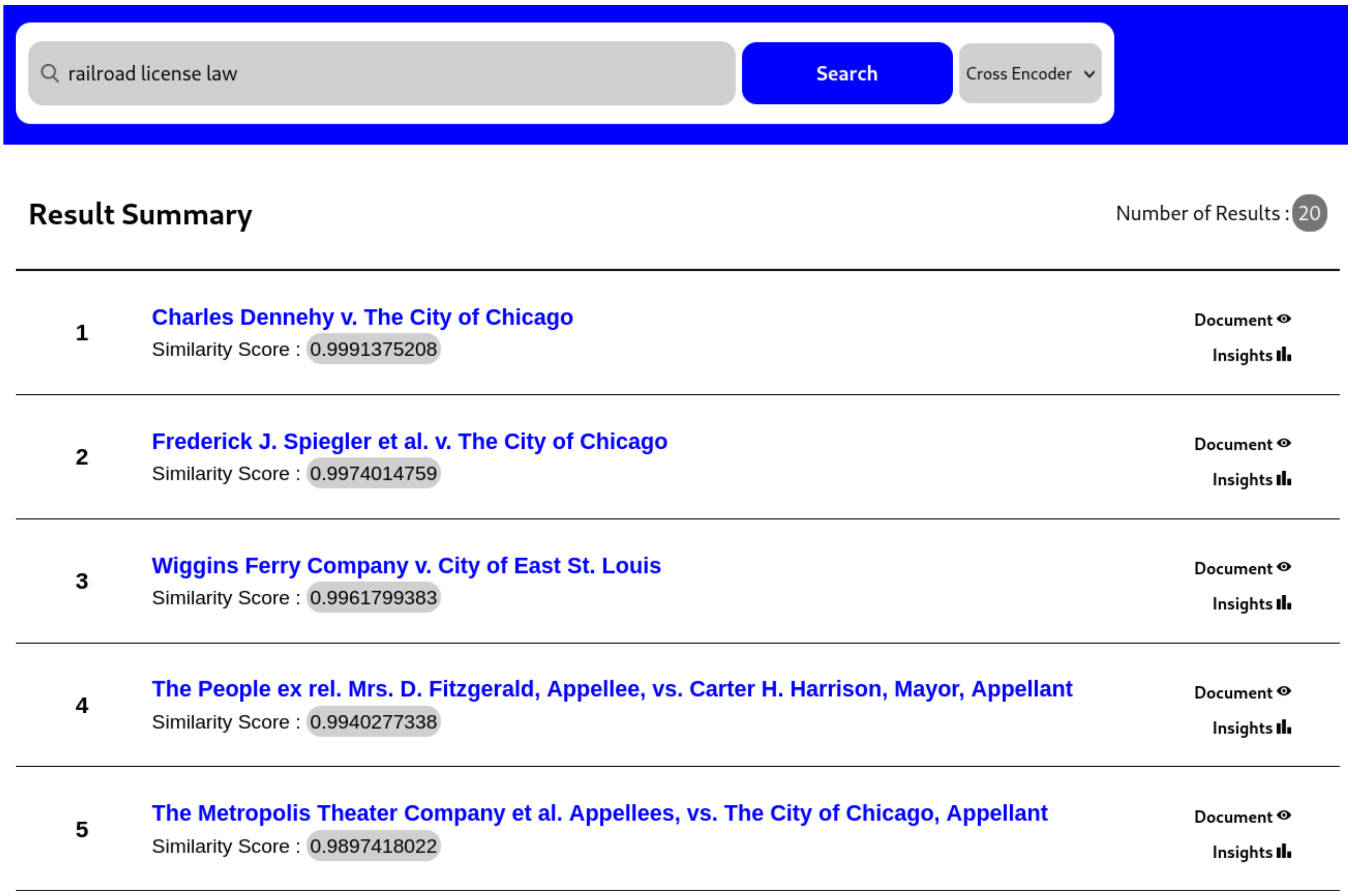
As opposed to the cross-encoder, the dual encoder tokenizes, ingests, and encode the query and top K most lexically relevant documents independently, producing separate vector representations for each. With this, the encodings for each document and query pair are generated in isolation without context across the two sequences. The encodings are produced based on the top encoding layer of the Transformer architecture, capturing the highest-level of semantic information of the sequences (whereas lower layers capture more grammatical and syntactical information) [9, 2, 29]. The query vector representation is compared against each document representation to produce a cosine similarity score.

$$\cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{4.1}$$

Figure 4.3: The formula for calculating cosine similarity between two vectors **A** and **B**.

The documents are then re-ranked according to this similarity score to produce the final ordering of results, ranked by semantic relevance.

4.2 User Interface Showcase



Result Summary

Number of Results : 20

1 Charles Dennehy v. The City of Chicago
Similarity Score : 0.9991375208
Document
Insights

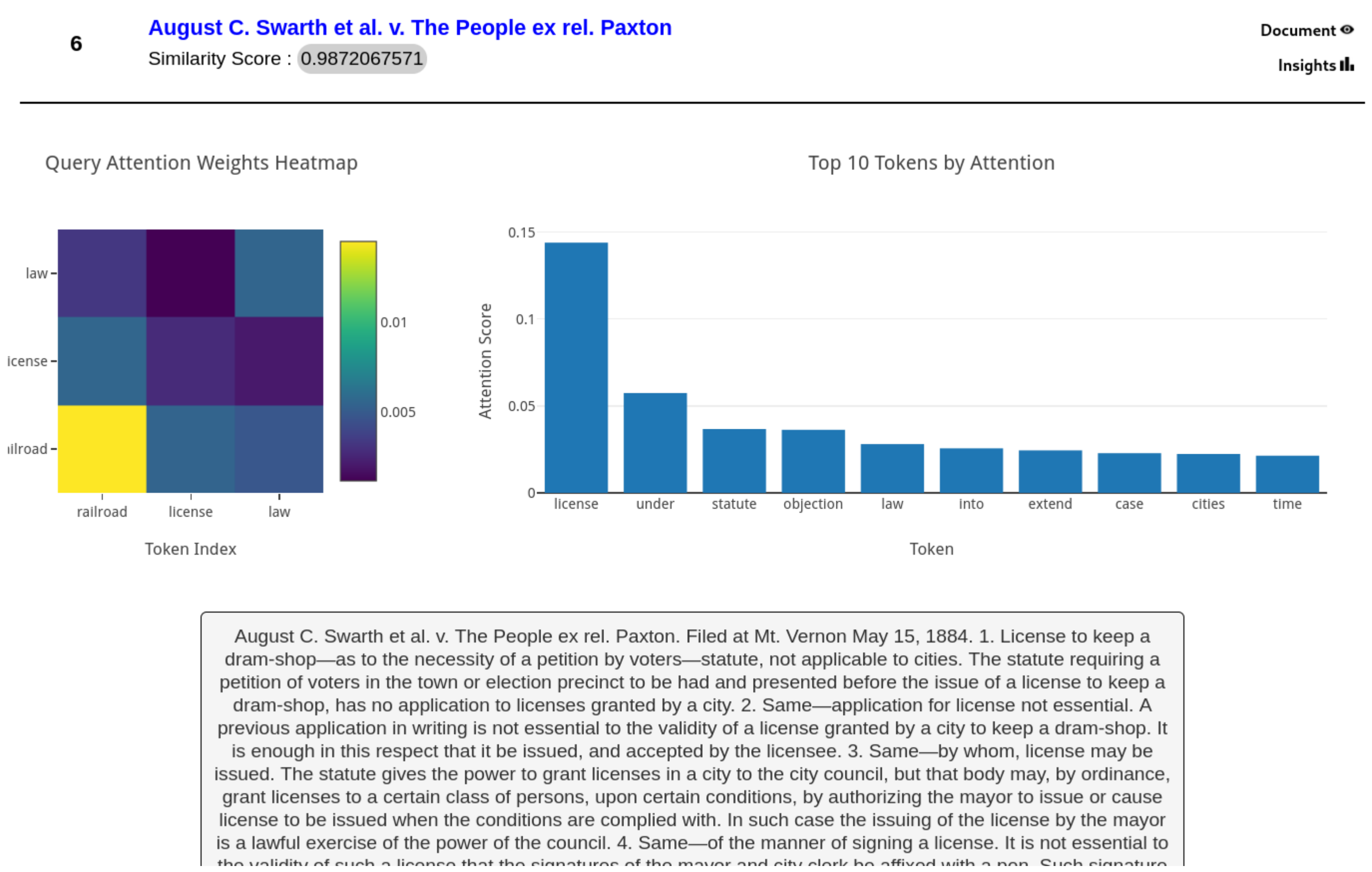
2 Frederick J. Spiegler et al. v. The City of Chicago
Similarity Score : 0.9974014759
Document
Insights

3 Wiggins Ferry Company v. City of East St. Louis
Similarity Score : 0.9961799383
Document
Insights

4 The People ex rel. Mrs. D. Fitzgerald, Appellee, vs. Carter H. Harrison, Mayor, Appellant
Similarity Score : 0.9940277338
Document
Insights

5 The Metropolis Theater Company et al. Appellees, vs. The City of Chicago, Appellant
Similarity Score : 0.9897418022
Document
Insights

Figure 4.4: Overview of user interface layout. We can see each result thumbnail contains the title of the document (court case in this instance) along with a similarity score computed by the deep learned encoder (cross-encoder in this instance)



August C. Swarth et al. v. The People ex rel. Paxton. Filed at Mt. Vernon May 15, 1884. 1. License to keep a dram-shop—as to the necessity of a petition by voters—statute, not applicable to cities. The statute requiring a petition of voters in the town or election precinct to be had and presented before the issue of a license to keep a dram-shop, has no application to licenses granted by a city. 2. Same—application for license not essential. A previous application in writing is not essential to the validity of a license granted by a city to keep a dram-shop. It is enough in this respect that it be issued, and accepted by the licensee. 3. Same—by whom, license may be issued. The statute gives the power to grant licenses in a city to the city council, but that body may, by ordinance, grant licenses to a certain class of persons, upon certain conditions, by authorizing the mayor to issue or cause license to be issued when the conditions are complied with. In such case the issuing of the license by the mayor is a lawful exercise of the power of the council. 4. Same—of the manner of signing a license. It is not essential to the validity of such a license that the signatures of the mayor and city clerk be affixed with a pen. Such signature

Figure 4.5: Clicking on the “document” and “insights” buttons on the expands the result row to display an Transformer attention heat-map for they query tokens and a bar chart showing which terms in the document received the most attention from the encoder. The attention values are extracted from the top encoding layer of the fine-tuned BERT transformer model. The top layer is typically convention for attention visualizations as it captures the highest level semantic understanding a sequence (where as lower layers capture more grammatical and syntactical information) [9, 2, 29].

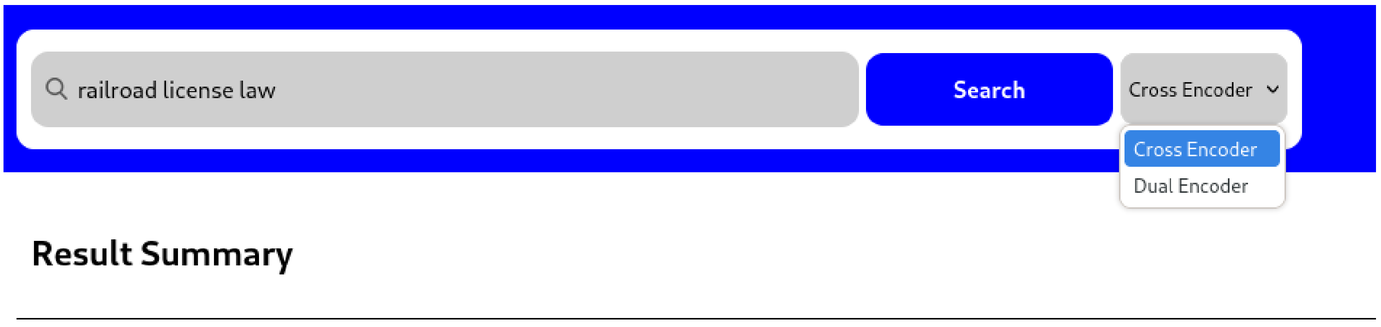


Figure 4.6: Drop-down for choosing between cross-encoder pipeline or bi-encoder pipeline

Chapter 5

Critical Appraisal & Conclusion

The nature of this project has always been exploratory from its conception. With respect to the original DOER document, this project was originally intended to primarily be a software engineering endeavour, where the end goal was centered around producing a fully functioning and deploy-able user application, with the auxiliary task of exploring the functionality and application of Transformer models. As the project has evolved, the focus has shifted towards the research component, in particular the fine-tuning and application of the Transformer model to increase performance and domain specificity for information retrieval in the legal domain. Consequently a new set of objectives were created to reflect this shift in focus. These objectives, and the extent to which they are completed are listed below.

5.0.1 Primary Objectives

1. Dataset Selection & Collection : Select a number of corpora and datasets suitable for the selected fine-tuning tasks and to serve as the corpus to perform information retrieval over. This objective includes implementing suitable data pre-processing steps.
2. Model Selection & Training Methodology : Select a suitable pre-trained transformer model for the domain and identify downstream tasks to use for fine-tuning the model for the legal domain and the task of information retrieval. This involves creating a number of fine-tuned models using a number of various fine-tuning strategies.
3. Bench marking & Evaluation : Benchmark and evaluate the performance of each fine-tuned model against each of the selected downstream tasks for fine-tuning as well as a the primary task of information retrieval.

These primary objectives have been met. A number of datasets have been selected for the purpose of implementing a number of fine-tuning objectives and a corpora of case document has been selected as an corpus to search over for the user application. The BERT architecture has been chosen as a Transformer architecture of choice and an investigation has been performed to evaluate a number of fine-tuning strategies that, while not exhaustive, explore a variety of fundamental fine-tuning paradigms within the field of language processing and information retrieval. In particular, these are domain specific pre-training, single-task fine-tuning, sequential multi-task fine-tuning, and parallel multi-task fine-tuning. In total, 5 models have been ultimately produced (with many other developmental models along the way). Using these 5 models, a number of considerations within information retrieval have been explored and discussed. In regards to bench-marking and evaluation, a number of evaluation methods have been implemented. The first method investigates the effect that each fine-tuning strategy has on the performance of BERT task-specific heads for sequence pair classification, a vital task for information retrieval. The second investigates how each fine-tuning strategy alters the parameters of the underlying transformer encoding layers, used to produce nuanced, context-aware representations suitable for downstream tasks.

5.0.2 Secondary Objectives

1. Implement Semantic Search: Utilize the fine-tuned model and an appropriate corpus for information retrieval. Develop a semantic search pipeline to process queries and deliver relevant results.

2. User Interface : Develop a user application around the core functionality of the fine-tuned model for information retrieval. This involves implementing a graphical user interface and backend surrounding the core information retrieval functionality. This is more of a secondary objective as to have more tangible visual demonstration of the search functionality and to explore common pipelines used for information retrieval.

The secondary objectives have been met. In particular, two pipelines have been implemented over a corpus of court case documents, each a hybrid combining multiple approaches for dense document/resource retrieval. The first approach combines a TF-IDF lexical engine with a fine-tuned cross encoder. The second approach combines a TF-IDF lexical engine with a fine-tuned bi-encoder. Finally a user interface was implemented along with a backend to connect the interface with the functionality of the two search pipelines.

5.1 Related Works

Overall this project serves to be a proof of concept. While the performance results are likely no where near that of industry or any published works, this project experiments and explores with several fundamental considerations salient in state of the art products and publications. Workings such as (Luo et al. (2023)[43], Liu et al. (2019)[40]) shed light on the potential utility (as well as possible downsides) of multi-task fine-tuning of transformer models over traditional single task approaches. This project finds the parallel multi-task fine-tuning approach did produce the highest performance metrics (for the given datasets) when compared to single task and sequential task variants. Working such as Gururangan et al. (2020)[24], Chaldikis et al. (2020)[5] introduce the utility of further pre-training, using masked language modelling, to increase domain specificity for the application of Transformer models for language tasks in specialized domains. The results of this project additionally align with these findings, where the combination of masked language model pre-training with parallel multi-task fine-tuning produced the highest accuracy for the textual entailment and selective question and answering tasks.

Bibliography

[1] . *1537. Manslaughter Defined.* www.justice.gov, Feb. 2015. URL: https://www.justice.gov/archives/jm/criminal-resource-manual-1537-manslaughter-defined.

[2] A. Vaswani et al. "Attention Is All You Need". In: (June 2017). DOI: 10.48550/arXiv.1706.03762. URL: https://arxiv.org/abs/1706.03762.

[3] Iz Beltagy, Arman Cohan, and Kyle Lo. "SciBERT: Pretrained Contextualized Embeddings for Scientific Text". In: *CoRR* abs/1903.10676 (2019). arXiv: 1903.10676. URL: http://arxiv.org/abs/1903.10676.

[4] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: *CoRR* abs/2005.14165 (2020). arXiv: 2005.14165. URL: https://arxiv.org/abs/2005.14165.

[5] Ilias Chalkidis et al. "LEGAL-BERT: The Muppets straight out of Law School". In: (2020). arXiv: 2010.02559 [cs.CL].

[6] Ilias Chalkidis et al. *Regulatory Compliance through Doc2Doc Information Retrieval: A case study in EU/UK legislation where text similarity has limitations.* 2021. arXiv: 2101.10726 [cs.CL].

[7] Manu Chatterjee. *Title Page for a Granted US Patent (image by author).* Apr. 1999. URL: https://miro.medium.com/v2/resize:fit:720/format:webp/0*eIrUxzS9Hs64rOKr.

[8] Marta Chroma. "Synonymy and Polysemy in Legal Terminology and Their Applications to Bilingual and Bijural Translation". In: *Research in Language* 9 (June 2011). DOI: 10.2478/v10015-011-0004-2.

[9] Kevin Clark et al. "What Does BERT Look at? An Analysis of BERT's Attention". In: (Aug. 2019). Ed. by Tal Linzen et al., pp. 276–286. DOI: 10.18653/v1/W19-4828. URL: https://aclanthology.org/W19-4828.

[10] *Consideration.* LII / Legal Information Institute. URL: https://www.law.cornell.edu/wex/consideration.

[11] *Consideration Definition | Legal Glossary | LexisNexis.* www.lexisnexis.co.uk. URL: https://www.lexisnexis.co.uk/legal/glossary/consideration#:~:text=Commercial-.

[12] *Cross-Encoders — Sentence-Transformers documentation.* www.sbert.net. URL: https://www.sbert.net/examples/applications/cross-encoder/README.html.

[13] *CrossEntropyLoss — PyTorch 1.6.0 documentation.* pytorch.org. URL: https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html.

[14] *Culpable Homicide.* Crime.Scot, Nov. 2018. URL: https://crime.scot/culpable-homicide/.

[15] *Data Collator.* huggingface.co. URL: https://huggingface.co/docs/transformers/en/main_classes/data_collator (visited on 03/19/2024).

[16] *Definition of RES JUDICATA*. Merriam-webster.com, 2020. URL: https://www.merriam-webster.com/dictionary/res%20judicata.

[17] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[18] Shachar Don-Yehiya et al. "ColD Fusion: Collaborative Descent for Distributed Multitask Finetuning". In: *acl anthology* 1 (2023), pp. 788–806. URL: https://aclanthology.org/2023.acl-long.46.pdf (visited on 03/25/2024).

[19] fdeloche. *Structure of RNN*. June 2017. URL: https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg.

[20] Filippo Galgani, Paul Compton, and Achim Hoffmann. "Combining Different Summarization Techniques for Legal Text". In: *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*. Ed. by Natalia Grabar et al. Avignon, France: Association for Computational Linguistics, Apr. 2012, pp. 115–123. URL: https://aclanthology.org/W12-0515.

[21] Ondřej Glogar. "The Concept of Legal Language: What Makes Legal Language 'Legal'?" In: *International Journal for the Semiotics of Law* (May 2023). DOI: 10.1007/s11196-023-10010-5.

[22] Ondrej Glogar. "The Concept of Legal Language: What Makes Legal Language 'Legal'?" In: *International Journal for the Semiotics of Law - Revue internationale de Sémiotique juridique* (May 2023). DOI: 10.1007/s11196-023-10010-5. URL: https://link.springer.com/article/10.1007/s11196-023-10010-5.

[23] Kunpeng Guo et al. "Fine-tuning Strategies for Domain Specific Question Answering under Low Annotation Budget Constraints". In: (Nov. 2023). DOI: 10.1109/ictai59109.2023.00032. URL: http://dx.doi.org/10.1109/ICTAI59109.2023.00032.

[24] Suchin Gururangan et al. "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks". In: *CoRR* abs/2004.10964 (2020). arXiv: 2004.10964. URL: https://arxiv.org/abs/2004.10964.

[25] Andrew Senior Hasim Sak and Francoise Beaufays. "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modelling". In: (2014). DOI: 10.48550/arXiv.1402.1128. URL: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf.

[26] Daniel J. Hemel. "Polysemy and the Law". In: *SSRN Electronic Journal* (2022). DOI: 10.2139/ssrn.4264800. (Visited on 03/22/2023).

[27] Tina Hershey and Donald Burke. "Pioneers in Computerized Legal Research: The Story of the Pittsburgh System". In: *Pittsburgh Journal of Technology Law and Policy* 18 (Feb. 2018). DOI: 10.5195/TLP.2018.212.

[28] Hongpeng Jin et al. *Rethinking Learning Rate Tuning in the Era of Large Language Models*. 2023. arXiv: 2309.08859 [cs.LG].

[29] Daniel Jurafsky and James H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Education, 2008. Chap. 11. ISBN: 9788131716724.

[30] Daniel Martin Katz et al. “Natural Language Processing in the Legal Domain”. In: *SSRN Electronic Journal* (2023). DOI: 10.2139/ssrn.4336224. URL: https://arxiv.org/pdf/2302.12039.pdf (visited on 06/30/2023).

[31] mi-young Kim et al. “COLIEE 2020: Methods for Legal Document Retrieval and Entailment”. In: (June 2021), pp. 196–210. DOI: 10.1007/978-3-030-79942-7_13.

[32] D. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *Computer Science* (2014). DOI: 10.48550/arXiv.1412.6980. URL: https://xueshu.baidu.com/usercenter/paper/show?paperid=37a73866f09edd03830b234716447e4f&site=xueshu_se.

[33] Anastassia Kornilova and Vladimir Eidelman. “BillSum: A Corpus for Automatic Summarization of US Legislation”. In: *Proceedings of the 2nd Workshop on New Frontiers in Summarization.* Ed. by Lu Wang et al. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 48–56. DOI: 10.18653/v1/D19-5406. URL: https://aclanthology.org/D19-5406.

[34] Abhishek Kumar and Hal Daume III au2. “Learning Task Grouping and Overlap in Multi-task Learning”. In: (2012). arXiv: 1206.6417 [cs.LG].

[35] Jinhyuk Lee et al. “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”. In: *CoRR* abs/1901.08746 (2019). arXiv: 1901.08746. URL: http://arxiv.org/abs/1901.08746.

[36] *Lexis+ AI | LexisNexis UK.* www.lexisnexis.co.uk. URL: https://www.lexisnexis.co.uk/lexis-plus/lexis-plus-ai.html (visited on 03/17/2024).

[37] Jonathan Li, Rohan Bhambhoria, and Xiaodan Zhu. “Parameter-Efficient Legal Domain Adaptation”. In: *Proceedings of the Natural Legal Language Processing Workshop 2022.* Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 119–129. URL: https://aclanthology.org/2022.nllp-1.10.

[38] Bingchang Liu et al. “MFTCoder: Boosting Code LLMs with Multitask Fine-Tuning”. In: (2023). arXiv: 2311.02303 [cs.LG].

[39] Xialei Liu et al. “Rotate your Networks: Better Weight Consolidation and Less Catastrophic Forgetting”. In: Aug. 2018. DOI: 10.1109/ICPR.2018.8545895.

[40] Xiaodong Liu et al. “Multi-Task Deep Neural Networks for Natural Language Understanding”. In: (2019). arXiv: 1901.11504 [cs.CL].

[41] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization.* 2019. arXiv: 1711.05101 [cs.LG].

[42] Adrienn Lukacs. “What is Privacy? The History and Definition of Privacy”. In: *core.ac.uk* (). URL: https://core.ac.uk/reader/80769180.

[43] Zhiyi Luo, Song Y Yan, and Shuyun Luo. “Multitask Fine Tuning on Pretrained Language Model for Retrieval-Based Question Answering in Automotive Domain”. In: *Mathematics* 11 (June 2023), pp. 2733–2733. DOI: 10.3390/math11122733. (Visited on 11/27/2023).

[44] Tamsin Maxwell and Burkhard Schafer. “Concept and Context in Legal Information Retrieval”. In: *Frontiers in Artificial Intelligence and Applications* 189 (Jan. 2008), pp. 63–72. DOI: 10.3233/978-1-58603-952-3-63.

[45] Michael McCloskey and Neal J. Cohen. "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem". In: Psychology of Learning and Motivation 24 (1989). Ed. by Gordon H. Bower, pp. 109–165. ISSN: 0079-7421. DOI: `https://doi.org/10.1016/S0079-7421(08)60536-8`. URL: `https://www.sciencedirect.com/science/article/pii/S0079742108605368`.

[46] *Models — transformers 3.0.2 documentation.* huggingface.co. URL: `https://huggingface.co/transformers/v3.0.2/main_classes/model.html` (visited on 03/20/2024).

[47] Pandu Nayak. *Understanding searches better than ever before.* Oct. 2019. URL: `https://blog.google/products/search/search-language-understanding-bert/`.

[48] David Newman. *Bag of Words.* UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C 2008.

[49] Dat T. Nguyen et al. "An Unsupervised Learning Method to improve Legal Document Retrieval task at ALQAC 2022". In: (2022), pp. 1–4. DOI: `10.1109/KSE56063.2022.9953618`.

[50] Hieu Nguyen Van et al. *Miko Team: Deep Learning Approach for Legal Question Answering in ALQAC 2022.* 2022. URL: `https://arxiv.org/pdf/2211.02200.pdf` (visited on 03/18/2024).

[51] *Notice.* The Free Dictionary. URL: `https://legal-dictionary.thefreedictionary.com/Notice`.

[52] Marc van Opijnen and Cristiana Santos. "On the concept of relevance in legal information retrieval". In: *Artificial Intelligence and Law* 25 (Mar. 2017), pp. 65–87. DOI: `10.1007/s10506-017-9195-8`.

[53] Jesus-German Ortiz-Barajas, Gemma Bel-Enguix, and Helena Gómez-Adorno. "Sentence-CROBI: A Simple Cross-Bi-Encoder-Based Neural Network Architecture for Paraphrase Identification". In: *Mathematics* 10.19 (2022). ISSN: 2227-7390. DOI: `10.3390/math10193578`. URL: `https://www.mdpi.com/2227-7390/10/19/3578`.

[54] *Practical BM25 - Part 2: The BM25 Algorithm and its Variables.* Elastic Blog, Apr. 2018. URL: `https://www.elastic.co/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables`.

[55] *PyTorch.* www.pytorch.org. URL: `https://pytorch.org/hub/huggingface_pytorch-transformers/`.

[56] Juan Ramos. "Using TF-IDF to determine word relevance in document queries". In: (Jan. 2003).

[57] Abhilasha Ravichander et al. "Question Answering for Privacy Policies: Combining Computational and Legal Perspectives". In: *CoRR* abs/1911.00841 (2019). arXiv: `1911.00841`. URL: `http://arxiv.org/abs/1911.00841`.

[58] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. "A Primer in BERTology: What we know about how BERT works". In: *CoRR* abs/2002.12327 (2020). arXiv: `2002.12327`. URL: `https://arxiv.org/abs/2002.12327`.

[59] Carlo Sansone and Giancarlo Sperlí. "Legal Information Retrieval systems: State-of-the-art and open issues". In: *Information Systems* 106 (2022), p. 101967. ISSN: 0306-4379. DOI: `https://doi.org/10.1016/j.is.2021.101967`. URL: `https://www.sciencedirect.com/science/article/pii/S0306437921001551`.

[60] Anton Maximilian Schaefer, Steffen Udluft, and Hans-Georg Zimmermann. “Learning long-term dependencies with recurrent neural networks”. In: *Neurocomputing* 71.13 (2008). Artificial Neural Networks (ICANN 2006) / Engineering of Intelligent Systems (ICEIS 2006), pp. 2481–2488. ISSN: 0925-2312. DOI: `https://doi.org/10.1016/j.neucom.2007.12.036`. URL: `https://www.sciencedirect.com/science/article/pii/S0925231208002105`.

[61] Keet Sugathadasa et al. “Legal Document Retrieval using Document Vector Embeddings and Deep Learning”. In: *CoRR* abs/1805.10685 (2018). arXiv: `1805.10685`. URL: `http://arxiv.org/abs/1805.10685`.

[62] Chi Sun et al. “How to Fine-Tune BERT for Text Classification?” In: (2019). Ed. by Maosong Sun et al., pp. 194–206.

[63] Nguyen Ha Thanh et al. “A Summary of the ALQAC 2021 Competition”. In: *2021 13th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, Nov. 2021. DOI: `10.1109/kse53942.2021.9648724`. URL: `http://dx.doi.org/10.1109/kse53942.2021.9648724`.

[64] P.M. Tiersma. *Legal Language*. University of Chicago Press, 1999. ISBN: 9780226803029. URL: `https://books.google.co.uk/books?id=Sq8XXTo3A48C`.

[65] Don Tuggener et al. “LEDGAR: A Large-Scale Multi-label Corpus for Text Classification of Legal Provisions in Contracts”. English. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 1235–1241. ISBN: 979-10-95546-34-4. URL: `https://aclanthology.org/2020.lrec-1.155`.

[66] Howard Turtle. “Text retrieval in the legal world”. In: *Artificial Intelligence and Law* 3.1 (Mar. 1995), pp. 5–54. ISSN: 1572-8382. DOI: `10.1007/BF00877694`. URL: `https://doi.org/10.1007/BF00877694`.

[67] *Using the West Key Number System*. legal.thomsonreuters.com. URL: `https://legal.thomsonreuters.com/en/insights/articles/using-the-west-key-numbers-system` (visited on 03/18/2024).

[68] Enrique Alcaraz Varoo and Brian Hughes. *Legal translation explained*. Routledge, 2015.

[69] Samuel D. Warren and Louis D. Brandeis. “The Right to Privacy”. In: *Harvard Law Review* 4.5 (1890), pp. 193–220. ISSN: 0017811X. URL: `http://www.jstor.org/stable/1321160` (visited on 03/16/2024).

[70] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. “A survey of transfer learning”. In: *Journal of Big Data* 3 (May 2016). DOI: `10.1186/s40537-016-0043-6`. (Visited on 03/06/2019).

[71] *Westlaw – Legal Research Tools & Platforms*. legal.thomsonreuters.com. URL: `https://legal.thomsonreuters.com/en/westlaw`.

[72] *Westlaw Edge - Advanced Legal Research powered by A.I.* thomsonreuters.com, Nov. 2021. URL: `https://legal.thomsonreuters.com/en/products/westlaw-edge`.

[73] WIPO. *PATENT CLAIM FORMAT AND TYPES OF CLAIMS*. URL: `https://www.wipo.int/edocs/mdocs/aspac/en/wipo_ip_phl_16/wipo_ip_phl_16_t5.pdf`.

[74] Wen Zhang et al. “A survey on negative transfer”. In: *IEEE/CAA Journal of Automatica Sinica* (2022), pp. 1–25. DOI: `10.1109/jas.2022.106004`. URL: `https://arxiv.org/pdf/2009.00909.pdf` (visited on 11/15/2022).

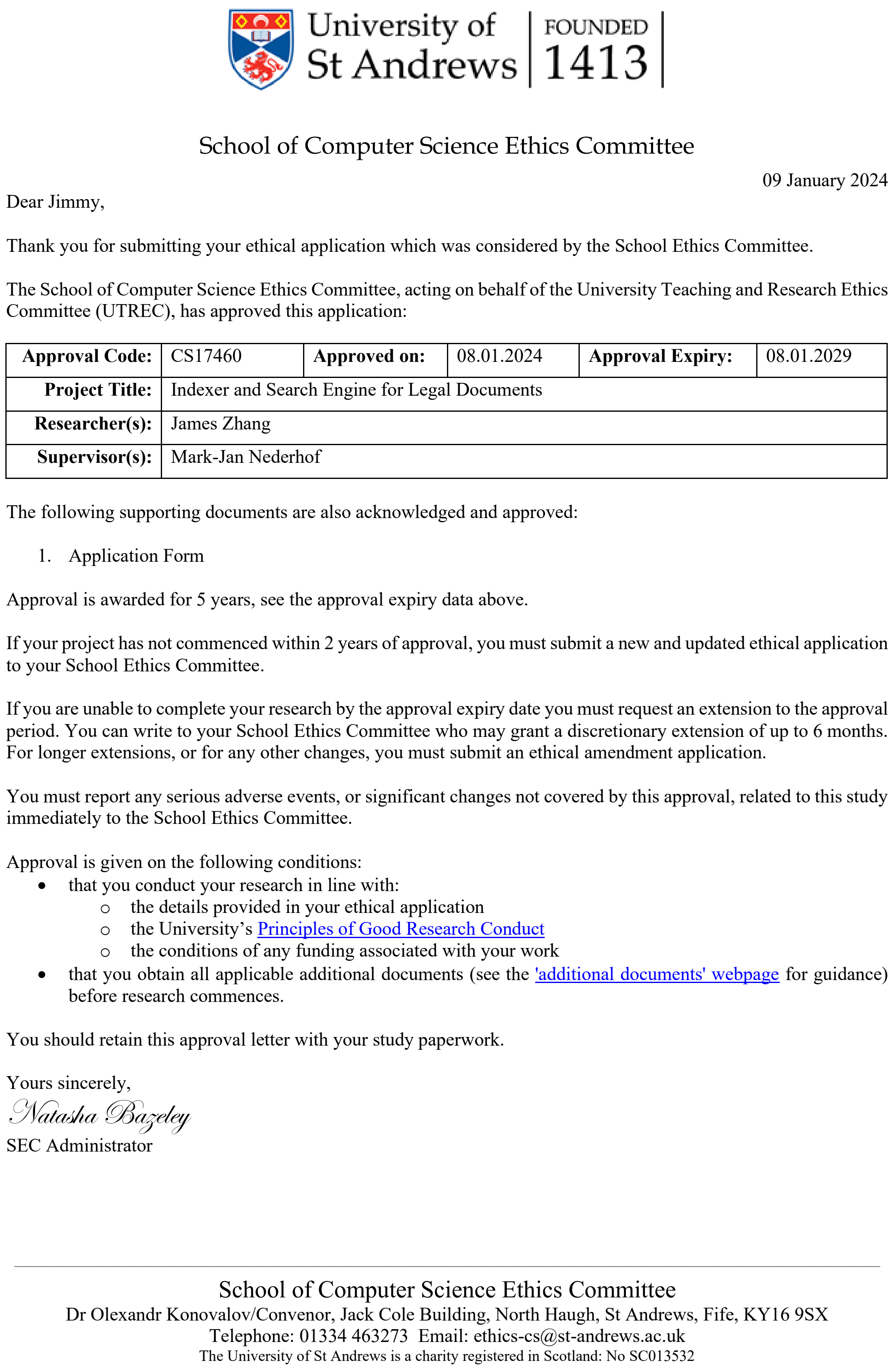
[75] Yiyun Zhao and Steven Bethard. “How does BERT’s attention change when you fine-tune? An analysis methodology and a case study in negation scope”. In: (July 2020). Ed. by Dan Jurafsky et al., pp. 4729–4747. DOI: 10.18653/v1/2020.acl-main.429. URL: https://aclanthology.org/2020.acl-main.429.

[76] Lucia Zheng et al. “When Does Pretraining Help? Assessing Self-Supervised Learning for Law and the CaseHOLD Dataset”. In: *CoRR* abs/2104.08671 (2021). arXiv: 2104.08671. URL: https://arxiv.org/abs/2104.08671.

Chapter 6

Appendix

6.1 Ethics Approval



University of St Andrews | FOUNDED 1413

School of Computer Science Ethics Committee

09 January 2024

Dear Jimmy,

Thank you for submitting your ethical application which was considered by the School Ethics Committee.

The School of Computer Science Ethics Committee, acting on behalf of the University Teaching and Research Ethics Committee (UTREC), has approved this application:

Approval Code:	CS17460	**Approved on:**	08.01.2024	**Approval Expiry:**	08.01.2029
Project Title:	Indexer and Search Engine for Legal Documents				
Researcher(s):	James Zhang				
Supervisor(s):	Mark-Jan Nederhof				

The following supporting documents are also acknowledged and approved:

1. Application Form

Approval is awarded for 5 years, see the approval expiry data above.

If your project has not commenced within 2 years of approval, you must submit a new and updated ethical application to your School Ethics Committee.

If you are unable to complete your research by the approval expiry date you must request an extension to the approval period. You can write to your School Ethics Committee who may grant a discretionary extension of up to 6 months. For longer extensions, or for any other changes, you must submit an ethical amendment application.

You must report any serious adverse events, or significant changes not covered by this approval, related to this study immediately to the School Ethics Committee.

Approval is given on the following conditions:

- that you conduct your research in line with:
  - the details provided in your ethical application
  - the University's Principles of Good Research Conduct
  - the conditions of any funding associated with your work
- that you obtain all applicable additional documents (see the 'additional documents' webpage for guidance) before research commences.

You should retain this approval letter with your study paperwork.

Yours sincerely,

Natasha Bazeley

SEC Administrator

School of Computer Science Ethics Committee
Dr Olexandr Konovalov/Convenor, Jack Cole Building, North Haugh, St Andrews, Fife, KY16 9SX
Telephone: 01334 463273 Email: ethics-cs@st-andrews.ac.uk
The University of St Andrews is a charity registered in Scotland: No SC013532