Criticality Prediction and Information Retrieval in Swiss Legal Data

Bachelor Thesis

Ronja Stern

from

Bern, Switzerland

Faculty of Science, University of Bern

16. Juni 2023

Supervisors:
PD. Dr. Matthias Stürmer
Joël Niklaus
Research Group for Digital Sustainability
Institute of Computer Science
University of Bern, Switzerland

Abstract

We introduce two novel tasks for legal Natural Language Processing (NLP), aiming to add diversity to the field. As certain cases may have a more significant impact on jurisdiction than others, legal experts are interested in predicting the level of controversy surrounding a case. Additionally, being able to retrieve relevant laws or Leading Decision (BGE) a case depends on could greatly reduce court costs. The potential use cases for both tasks are numerous and not only include reducing court delays but also prioritizing cases for better judgments. To our knowledge, there have been no previous attempts to predict the criticality of a legal case or the retrieve relevant laws and BGE in Switzerland.

To address this, we publicly release two multilingual datasets containing cases from the Federal Supreme Court of Switzerland. The Swiss-Criticality-Prediction dataset features two approaches for labeling cases as critical, while the Swiss-Doc2doc-IR dataset includes links for each case to cited laws and BGE. In the Criticality task, we assess the performance of multilingual BERT-based models. For the Information Retrieval task, we evaluate existing models from the BEIR benchmark. We observe that all models fall short in their performance. Nonetheless, we find that domain-specific pre-training proves to be advantageous for both tasks.

Acknowledgments

I would like to express my gratitude to my supervisors PD. Dr. Stürmer and Joel Niklaus from the Research Center for Digital Sustainability for their input, guidance, and support throughout the duration of my Bachelor's thesis. Their encouragement, patience, and assistance have been crucial in the successful completion of this work. I would also like to extend my thanks to my fellow student, Visu, for his unwavering sense of humor and his ability to handle things in a light-hearted manner. Furthermore, I would like to express my appreciation to all of my fellow students from the NLP seminar for their ongoing support, and I wish them great success in the completion of their theses. Lastly, I am deeply grateful to my parents and friends for their unwavering support throughout this journey. Their belief in me and their critical proofreading of this thesis have been of immense help.

Contents

1 Introduction **1**

2 Background **2**
- 2.1 Introduction to Natural Language Processing . . . 2
  - 2.1.1 Language Representation . . . 2
  - 2.1.2 Transfer Learning . . . 3
- 2.2 Language Models . . . 3
  - 2.2.1 Transformers . . . 3
  - 2.2.2 BERT . . . 4
  - 2.2.3 Hierarchical Models . . . 5
  - 2.2.4 Bi-Encoder and Cross Encoder . . . 5
- 2.3 Introduction to Information Retrieval . . . 6
  - 2.3.1 Term Frequency - Inverse Document Frequency . . . 6
  - 2.3.2 BM25 . . . 6
  - 2.3.3 Re-ranking . . . 7
  - 2.3.4 Dense Passage Retrieval . . . 7
  - 2.3.5 Other . . . 7
- 2.4 Metrics . . . 8
- 2.5 Organisation Swiss Federal Supreme Court . . . 9
  - 2.5.1 Publications of Supreme Court Decisions . . . 9
  - 2.5.2 Sections . . . 10

3 Related Work **11**

4 Methods and Approach **13**
- 4.1 Pipeline . . . 13
- 4.2 Criticality Prediction . . . 14
  - 4.2.1 Criticality Dataset Creator . . . 14
  - 4.2.2 Task Configurations . . . 16
  - 4.2.3 Experiment Set Up . . . 17
- 4.3 Information Retrieval . . . 17
  - 4.3.1 Doc2doc Dataset Creator . . . 17
  - 4.3.2 Structure Data into Query, Corpus, and Qrel . . . 18
  - 4.3.3 Experiment Set Up . . . 19

5 Datasets **21**
- 5.1 Swiss-Criticality-Prediction Dataset . . . 22
  - 5.1.1 Sources of Errors . . . 22
- 5.2 Swiss-Doc2doc-IR Dataset . . . 23

6 Experiments **25**
6.1 Criticality . . . 25
6.1.1 Hyperparameters . . . 25
6.1.2 Results . . . 26
6.1.3 Discussion . . . 26
6.2 Information Retrieval . . . 26
6.2.1 Results . . . 26
6.2.2 Discussion . . . 27

7 Conclusion **29**

A SCALE **30**

B Dataset Distributions Swiss-Criticality-Prediction **31**

C Dataset Distributions Swiss-Doc2doc-IR **34**

List of Figures

2.1 Tokenization of a sentence with BERT tokenizer 2
2.2 General architecture of a Transformer 3
2.3 General architecture of a Bidirectional Encoder Representations from Transformers (BERT)-based model 4
2.4 Architecture of a single encoder 4
2.5 Comparison of Bi-Encoders like S-BERT and Cross Encoders 5

4.1 Workflow Overview 13
4.2 Pipleline of scraping Swiss Legal Documents 14
4.3 Relation of Federal Supreme Court Decisions (FSCD) and BGE considering labels 15
4.4 Method to create ranking of BGE 16
4.5 Process citations found in FSCD 18
4.6 Structure of corpus, queries and qrels 18

5.1 FSCD distribution over the years 21
5.2 BGE-label distribution 22
5.3 Citation-label distribution 22
5.4 Different steps to find FSCD for BGE cases 23
5.5 References found in BGE headers but not found as FSCD 23

6.1 Example for parameters 25

A.1 SCALE 30

B.1 Section facts input length distribution 31
B.2 Section considerations input length distribution 31
B.3 Section rulings input length distribution 32
B.4 BGE citation scores before and after weighting 32
B.5 Zooming into weighted BGE citation scores 32
B.6 Extracted references in BGE header per year 33

C.1 Section facts input length distribution 34
C.2 Laws and BGE citation amount distribution 34

List of Tables

4.1 Overview of the specifications of the language model 17

5.1 HuggingFace Datasets Overview 21
5.2 Criticality Task Configurations 22
5.3 Distribution of the number of laws and BGE citations in FSCD 24
5.4 Comparison of queries and corpus length distribution of different IR datasets . . 24

6.1 Criticality Prediction main results 26
6.2 Results IR: using a subsets of 100 queries 27
6.3 Results IR: using dataset adaptions 27

Abbreviations

BERT	Bidirectional Encoder Representations from Transformers
BGE	Leading Decision
BOW	Bag of Words
DCG	Discounted Cumulative Gain
DF	Document Frequency
DPR	Dense Passage Retrieval
doc2doc	Document-to-Document
FN	False Negative
FP	False Positive
FSCD	Federal Supreme Court Decisions
FSCS	Swiss Federal Supreme Court
IDF	Invers Document Frequency
IR	Information Retrieval
MCC	Matthews Correlation Coefficient
MLIR	Multilingual Information Retrieval
NDCG	Normalized Discounted Cumulative Gain
NLP	Natural Language Processing
S-BERT	Sentence-BERT
SLCT	Single Label Classification Task
SotA	state-of-the-art
TF	Term Frequency
TF-IDF	Term Frequency - Inverse Document Frequency
TN	True Negative
TP	True Positive

Chapter 1

Introduction

With the growing success of language models [20], the potential for their use in the field of law is becoming increasingly apparent [17, 22, 29, 37]. The legal field is known for its reliance on an enormous amount of text documents, written in a complex language with long and intricate sentences [13]. This makes it difficult for even experienced lawyers to keep all the details in mind. However, language models surpass human memorization by far. The challenge for these models lies in their ability to understand the contents and complex relationships within these texts.

In this thesis, we aim to introduce a new legal text classification task and a multilingual information retrieval task. The goal is to predict the criticality of a Swiss Federal Court decision and to predict the law articles or leading decisions that will be cited in the case. We will build on the existing Swiss Judgment Prediction task [28] and use its results as a baseline for our new tasks. Moreover, these tasks can also help in improving legal research as they can assist lawyers and legal experts in finding relevant cases and laws more efficiently. Additionally, they can provide insight into the development of jurisprudence and help identify trends and patterns in legal decision making. In the future, these models could also be used to automate the process of legal research and context analysis, freeing up valuable time and resources for legal experts. They have the potential to significantly reduce the cost of legal services and make the legal system more accessible and efficient. In conclusion, if models are found to perform those tasks well enough, the legal text classification and information retrieval tasks introduced in this thesis have the potential to bring numerous benefits to legal professionals and the wider society.

This initial attempt, to introduce these new and intriguing tasks, is not expected to produce outstanding results. Rather, the aim is to demonstrate that current state-of-the-art (SotA) models are not capable of delivering satisfactory outcomes (yet). As language models continue to advance, it is important to also push their limits and test their capabilities in new and challenging domains. This is especially important in areas like law where the use of complex and lengthy texts is common.

The tasks of this thesis is part of the SCALE Benchmark introduced by Rasiah et al. [31]. Before diving into the heart of our subject matter, we will start with a short overview of the key techniques and models important for our study. Following that, we will shed light on our methods and prior practices. Subsequently, delve into quantitative details in the chapters dedicated to datasets and experiments.

Chapter 2

Background

2.1 Introduction to Natural Language Processing

Natural Language Processing (NLP) is a multidisciplinary field that combines linguistics and machine learning with the aim of enabling computers to not only understand individual words, but also the context in which they are used, and to generate and classify human speech. However, since computers process information differently than humans, this is a highly challenging task. To tackle this challenge, machine learning and deep learning algorithms are employed as tools for NLP as language models. Machine learning algorithms are designed to detect complex patterns and relationships in data, with the intention of *learning* from the data without being explicitly programmed. In this work, we will be focusing on supervised learning approaches, where the label to be predicted is provided. Deep learning is an advanced and sophisticated subset of machine learning that utilizes neural networks to model and solve complex hierarchical representations of data. While it has demonstrated impressive performance in tasks such as image recognition, speech recognition, and NLP, it also requires significantly more computational resources and data for training than traditional machine learning methods. Nonetheless, the potential applications of deep learning algorithms are vast and continue to expand as research in this area advances [23].

2.1.1 Language Representation

```
sentence = "The quick brown fox jumps over the lazy dog."

tokens = ["the", "quick", "brown", "fox", "jump", "s", "over", "lazy", "dog", "."]

token2int = { "[PAS]": 0, "[UNK]": 1, "[CLS]": 2, "[SEP]": 3, "[MASK]": 4, "the":  5, "quick":  6, "brown": 7,
              "fox":  8, "jump":  9, "s":  10, "over":  11, "lazy":  12, "dog":  13, ".": 14}

tensor = [2, 5, 6, 7, 8, 9, 10, 11, 5, 12, 13, 14, 0, 0, 0, 0]
attention_mask = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
```

Figure 2.1. Tokenization of a sentence with BERT tokenizer

In order to process language, it needs to be considered how to represent text data. There are many different ways to represent text, such as Word2Vec or Bag of Words (BOW) models [27]. Those methods rely on a statistical representation of text. For optimal performance, we require a representation method that can capture the context and meaning of entire phrases, not just individual words. This is where tokenizers come into play [18]. A tokenizer breaks sentences down into tokens, which can be words, subwords, or special characters. These tokens are then assigned a unique integer value (illustrated as 'token2int' in Figure 2.1). Further the tokenizer

may introduce additional tokens into the sequence to facilitate the operation of the language model. In case of a classification task that leverages a BERT-based model (see Section 2.2.2) the special token [CLS] is appended to represent the class that is to be predicted subsequently. Besides, there are other special tokens always present by default which will not be detailed here. This process creates a sequence of integers that can be fed into the model for processing. Models may require a predetermined input length, which is why the sequence is padded with zeros to a fixed length. To indicate to the model that the padded zeros are not relevant, an attention mask is employed. For each token, the attention mask indicates whether it is relevant (1) or not (0) resulting in two tensors of the same length. See example in Figure 2.1.

2.1.2 Transfer Learning

In principle, when utilizing a neural network, our goal is to identify the optimal weights or parameters. These weights are adjusted in every training iteration to enhance the overall performance of the model. Models can be trained from scratch, implying that we start with randomly initialized weights. Given that training is computationally intensive and can span over several weeks, it is beneficial to use weights from a pre-trained model [42]. After pre-training, in the context of NLP, a model attains a statistical comprehension of language, but may not be able to tackle specific tasks. To address this, researchers fine-tune the pre-trained model for a task-specific application [36]. Typically, labeled data is used in this stage to guide the model's adjustments towards the desired target. This dual-phase procedure of pre-training and fine-tuning a model is referred to as transfer learning.

2.2 Language Models

In this section language models used in this thesis are presented.

2.2.1 Transformers

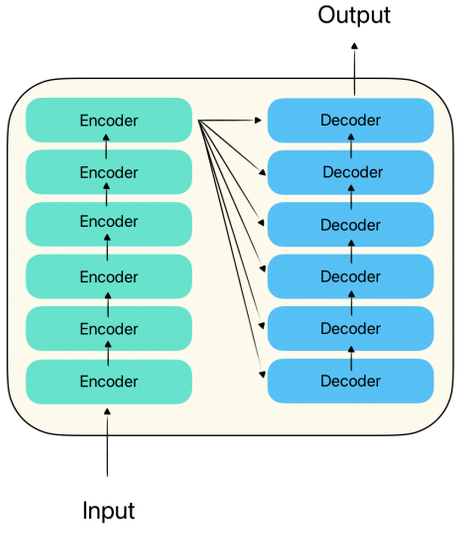

Figure 2.2. General architecture of a Transformer

In this thesis all language models are based on the transformers architecture introduced by Vaswani et al. [40]. Transformers have become one of the most promising model architectures for NLP. The original introduced Transformers architecture consists of 6 layers of encoders and 6 layers of decoders as shown in Figure 2.2. The encoder takes an input and builds a representation of its features, optimizing it to represent and understand the input. The decoder, on

the other hand, uses the output of the encoder, along with other inputs, to generate a target sequence. Depending on the task a Transformer architecture can be adapted. For text classification, an encoder-only architecture has advantages. This is due to our focus on representing the input for classification purposes, rather than creating a new output, a task for which a decoder architecture would be more suitable. Popular examples of language models using the Transformers architecture include BERT, GPT, and BART.

2.2.2 BERT

Bidirectional Encoder Representations from Transformers (BERT) [10] serves as the foundation for current state-of-the-art language models, particularly for classification tasks, due to its encoder-only architecture. As its name suggests, it uses only encoders. The Base model includes 12 encoder layers, while large models can have up to 24 encoder layers. The number of layers in a model impacts the number of parameters in the model, which in turn affects the computational resources required for using the model. As the number of layers increases, the model's parameter count grows, resulting in higher computational requirements. Figure 2.3 depicts the general architecture of a BERT-based model. A possible example of model input can be represented using tokens (t) and attention mask (am) as described in Section 2.1.1. It is important to note that all vectors, including tokens, attention masks, and representations, have the same length denoted by "n," which is defined by the specific model being used. To adapt BERT for specific tasks, a model head is attached to the encoder layers. The model head is designed specifically for the task at hand, such as binary classification. It transforms the representation vector r obtained from the encoders into a binary output, typically 0 or 1, corresponding to the classes involved in the classification task.

Figure 2.4 illustrates the structure of a single encoder, which again consists of various layers. One of the key advancements of BERT-based models is the Multi-Head Attention layer. This attention mechanism plays a crucial role in helping the model understand the relationships and meanings between words. By calculating the importance of each word and focusing on the more important ones, the attention mechanism improves the model's ability to represent human language effectively [9].

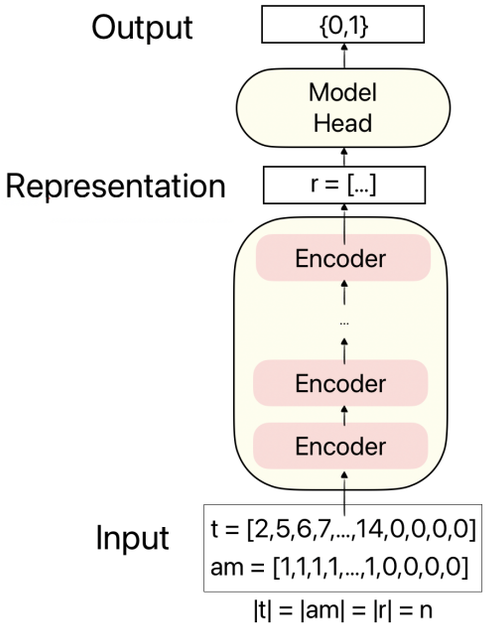

Figure 2.3. General architecture of a BERT-based model

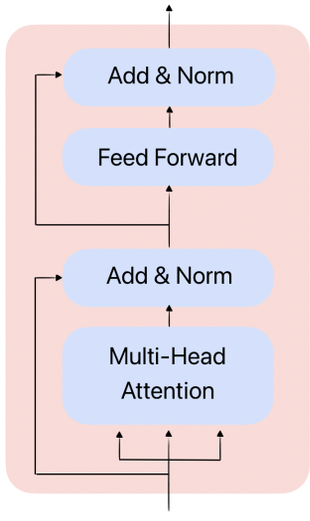

Figure 2.4. Architecture of a single encoder

BERT has achieved impressive results in various NLP tasks, including the GLUE benchmark.[1] Further improvement could be achieved, as demonstrated by Liu et al. [24] using an alternative pre-training objective. With training on multilingual data (including German, French and Italian) multilingual models like XLM-Roberta were published [6]. To account for the structural differences between legal language and normal language, it may be advantageous to use models that are additionally pre-trained on legal data.

2.2.3 Hierarchical Models

Language models like BERT are limited in their maximum input length. To enable longer inputs than the normal 512 tokens, hierarchical models have been introduced, such as the one used by Chalkidis et al. [3]. The input is split into segments of 512 tokens. In the first step, the BERT encoder processes each segment of text and encodes each segment independently. Since all segments should be combined, an additional "Bi-directional Long Short-Term Memory" (BiLSTM) encoder is used to aggregate the encodings from each segment. The final output states from the BiLSTM are concatenated to create a single representation of the entire document, which can then be used for classification purposes.

2.2.4 Bi-Encoder and Cross Encoder

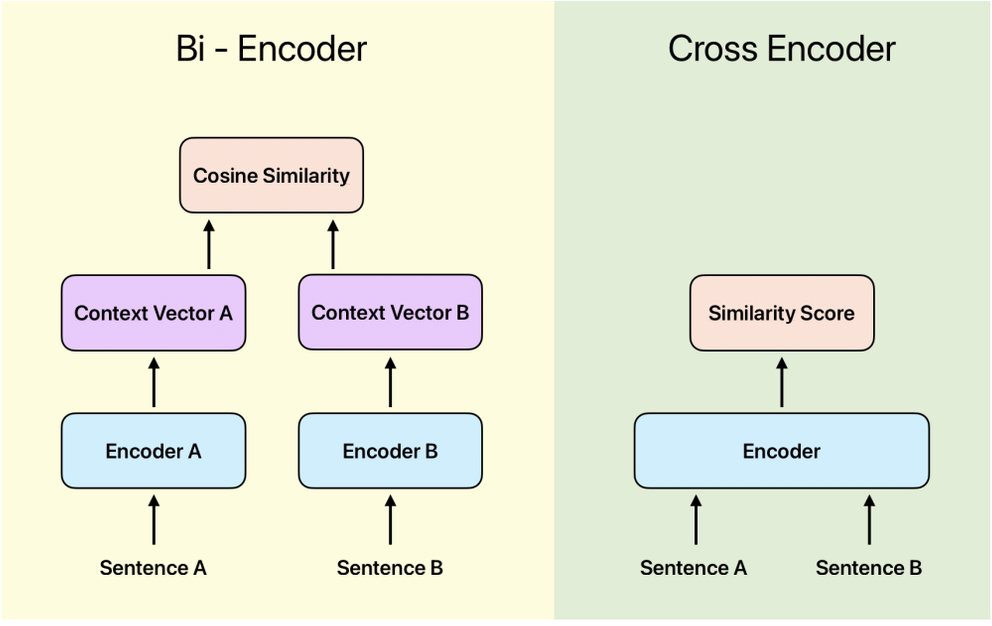

Figure 2.5. Comparison of Bi-Encoders like S-BERT and Cross Encoders

Bi-Encoder and Cross Encoder serve as additional examples of adapted BERT model architectures. The Bi-Encoder treats two inputs in parallel, utilizing separate encoders for each input. Each encoder independently transforms the given input into a contextual vector. These two vectors are then compared using either Dot Product or Cosine Similarity (in our example Cosine Similarity is used). One example of such a model is Sentence-BERT (S-BERT) [32] where two separate BERT models are used. On the contrary, the Cross Encoder combines both inputs and encodes them into a single vector representation. This single vector representation allows the calculation of a similarity score between the initial sentence inputs A and B. Cross Encoders can be computationally expensive since they need to compute each combination of the two inputs. However, they have shown state-of-the-art performance in re-ranking tasks [37].

[1] https://gluebenchmark.com/

2.3 Introduction to Information Retrieval

Information Retrieval (IR) can be likened to a modern-day "treasure hunt", with the aim of discovering the most pertinent information from an extensive and often overwhelming collection of documents for a given input. In this context, the input is called a **query**, while the assemblage of all information-containing documents is known as the **corpus**. A query-corpus pair is termed a **qrel**. The challenge of "finding a needle in a haystack" pales in comparison to the sheer volume of information available online. IR tackles this problem through the use of advanced algorithms and techniques to efficiently navigate through vast amounts of data and deliver the most relevant results. A web search engine serves as a prime example of the power of IR in our daily lives. The field of IR is constantly evolving, adapting to the ever-expanding universe of digital content, and utilizing diverse approaches to identify and retrieve relevant information. For a prolonged period, the most effective models were those representing the terms present in a given document. These models are based on the concept of Term Frequency (TF), which calculates the frequency of occurrence of a term in a document. Additionally, Document Frequency (DF) is utilized to count the number of documents in a collection where a term appears. Subsequently, we present two models utilizing TF and DF, respectively.

$$TF(t,d) = \text{number of occurances of term t in document d}$$
$$DF(t) = \text{number of documents where term t occurs}$$

2.3.1 Term Frequency - Inverse Document Frequency

Term Frequency - Inverse Document Frequency (TF-IDF) is a widely used retrieval method that operates on the basis of terms. It involves comparing vector representations of documents to an input query, with the closest matching vector being considered the most relevant. To facilitate this process, each document must first be represented as a vector. This is achieved through the use of the BOW model, which represents a document based on the TF and the Invers Document Frequency (IDF) of each term. Additionally one can use the log of TF instead of the TF.

One of the issues with using the BOW model is that it treats all terms equally, without considering their importance in the overall collection of documents. To address this, the concept of DF is used, which counts how many documents a term appears in. Terms that appear frequently in the collection are considered less informative and are given a lower weight. This is captured by the adapted IDF score, which is calculated as the logarithm of the total number of documents divided by the document frequency of the term. By multiplying the TF and IDF scores, we arrive at the TF-IDF score, which gives higher weight to terms that are more relevant to a specific document and less weight to common terms that do not add much meaning. Find the calculation of the TF-IDF score depending on term t, document d and the number of documents n below.

$$IDF(t) = log(\frac{n}{DF(t)})$$
$$TF\text{-}IDF(t,d) = TF(t,d) * IDF(t)$$

2.3.2 BM25

BM25 is an improved retrieval method that considers the TFs and takes into account the saturation effect and document length [33]. The saturation effect refers to the point where the relevance of a term stops increasing, even if it appears many times in a document. To address this issue, BM25 uses a parameter called "k" to control how quickly the relevance score of a term will saturate. Document length is another factor that BM25 takes into account. Longer documents are more likely to contain a higher number of occurrences of a term simply because

they contain more words, not necessarily because the term is more relevant to the document. BM25 incorporates another parameter called "b" to address this issue, controlling how much weight the document length has on the relevance score. By tuning these two parameters, BM25 can better reflect the relevance of terms in longer documents and avoid overestimating the importance of frequently occurring terms. Let A be the fraction of the length of documents d with the average document length. The exact formula of BM25 is shown below where b and k are the BM25 specific parameters (also called Lucene parameters) and Q is the set of all queries.

$$BM25(d, Q, b, k) = \sum_{t \in Q} IDF(t) \frac{(k+1)TF(t,d)}{(1-b)(b*A)+TF(t,d)}$$

As a lot of progress was made recently in NLP, also in IR new models and techniques were introduced. The biggest downside of models using TF is their lack of understanding a text. Newer models try to take this into account. In the following we introduce Re-ranking and Dense models. While Re-ranking models still use models like BM25 in the first stage, Dense models are completely independent of TF bases models. One goal of the following models is to close the lexical gap. [1] A lexical gap refers to a situation where a language does not have a specific word or phrase to express a particular concept or idea. This means that speakers of that language have to use a circumlocution or borrow words from other languages to express the idea.

2.3.3 Re-ranking

Re-ranking models are primarily differentiated between the pre-fetching and re-ranking phases. The pre-fetching phase involves searching the document corpus using IR techniques such as TF-IDF or BM25, to retrieve the best m results. In the re-ranking phase, a more sophisticated model such as BERT, Bi-Encoder or Cross Encoder are utilized to calculate the similarity between the query and one of the retrieved results. The results are then re-ranked, with the document most similar to the query according to the re-ranker model appearing on top [35]. As an example Thakur et al. [37] achieved best results for their IR tasks using Cross Encoders.

2.3.4 Dense Passage Retrieval

Dense Passage Retrieval (DPR) models can capture semantic matches and map queries and documents in a shared dense vector space. One way to achieve this, is using Bi-Encoders. During training, DPR is provided with a question-query pair and optimizes its weights by using either dot product or cosine similarity, with the goal of maximizing the score between each context-query pair and minimizing it when the query and context vectors are not a match. This way the encoders are trained to produce similar vectors for related query-context pairs. To find relevant documents for a query, the query vectors generated by the first Encoder are compared with the context vectors generated by the second Encoder and stored in the document store.

2.3.5 Other

There are some more terms related to IR, which will be used. In **Multilingual Information Retrieval** (**MLIR**) documents and queries are written in different languages. The goal of MLIR is to enable finding information no matter in what language they are written as presented in [11]. One approach to enable this, is if search engines are using translation technology. Mainly it must to be considered what should be translated - the documents or the queries? Secondly it must be defined how text is broken down into terms which are translated. Document-to-Document (doc2doc) represents the approach to search relevant documents for a given document.

In contrast to other approaches this ends in processing and comparing long texts instead of a few sentences.

2.4 Metrics

Given the complexity and nuances of language, a variety of metrics exist to evaluate language models. These metrics assess different aspects of the model, such as grammar, context, creativity, and style. The relevance of a particular metric depends on the specific application of the language model, such as text classification or question answering. In text classification, metrics such as Accuracy, Precision, and Recall are commonly used. All of those metrics are using the so called confusion matrix, which contains the count of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

$$
\begin{aligned}
&Accuracy = TP + TN/(TP + TN + FP + FN)\\
&Precision = TP/(TP + FP)\\
&Recall = TP/(TP + FN)\\
&F1 - Score = 2 * Precision * Recall/(Recall + Precision)\\
&Macro - F1 - Score = ((F1 - Score)_1 + ... + (F1 - Score)_n)/n\\
&MCC = (TP * TN - FP * FN)/sqrt((TP + FP)(TP + FN)(TN + FP)(TN + FN))
\end{aligned}
$$

Precision is defined as the ratio of TP to the sum of TP and FP. It represents the ability of the classifier to correctly identify negative samples as such. The ideal Precision score is 1, while the lowest possible value is 0.

Recall calculates the proportion of positive samples that were correctly identified by the model. It is expressed as the number of TP divided by the sum of TP and FN. A high Recall score indicates that the model is effective at detecting all positive instances, but it does not guarantee the quality of its predictions for negative instances.

Macro F1 is a measure of the overall performance of a classifier on a multi-class single-label classification problem. It is the harmonic mean of the Precision and Recall for each class, and is calculated by taking the average of the F1 scores for each class. It is a more comprehensive measure of the classifier's performance than Recall or Precision only. However, it is important to note that the Macro F1 Score can be misleading in cases where the class distribution is imbalanced, as it gives equal weight to each class regardless of the number of instances in that class.

Matthews Correlation Coefficient (MCC) is a single value metric that summarizes the confusion matrix similar to the Macro F1. When working with highly imbalanced classes, MCC has advantages over Macro F1 Score. The MCC takes also TN into account. This makes MCC a comprehensive metric for evaluating the performance of a classifier in imbalanced classification problems.

Normalized Discounted Cumulative Gain (NDCG) is used to measure the effectiveness of an IR task. Highly relevant documents should appear earlier in the search engine results list, as they are more useful than marginally or non-relevant documents. The NDCG metric sums up the real rankings of documents in the search engine results, divides them with the log of their ranking in the found list and normalizes this score in a last step as seen in the formula below. Disadvantages of this metric is that bad documents in the result list are punished. Neither get missing relevant documents penalized. Calculation of the NDCG and the Discounted Cumulative Gain (DCG) score where r represents the ranking and N the amount of documents

found as relevant are shown below. $DCG_{perfect}$ indicates the DCG score if the perfect documents were retrieved.

$$DCG = r_1 + \sum_{i=2}^{N} r_i / log_2(i)$$
$$NDCG = DCG / (DCG_{perfect})$$

Capped Recall @k is a measure of the effectiveness of a document retrieval model in retrieving relevant documents within the top k extracted documents. It is calculated as the fraction of relevant documents for a query that are retrieved from a scored list of documents provided by the model. As results might be misleading as the score never reaches 1 for big k even if all retrieved documents were relevant, the Recall score is capped at k for datasets where the number of relevant documents for a query is greater than k.[2] This score does not consider rankings of found documents, which is an advantage for tasks where the order is not relevant.

$$R\_Cap@k = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{|max_k(A_i) \cap A_{i*}|}{(min(k, |A_{i*}|)}$$

2.5 Organisation Swiss Federal Supreme Court

As we will work with data from the Swiss Federal Supreme Court (FSCS) we will introduce the Swiss system in the follwong section. The FSCS is the highest judicial authority in Switzerland, serving as the final court of appeal for cases from the federal criminal court (Bundesstrafgericht), the federal administrative court (Bundeverwaltungsgericht), the federal patent court (Bundespatentgericht), and cantonal courts. Its decisions play a crucial role in shaping the development and evolution of the law. The Federal Supreme Court is divided into seven different divisions, primarily based on the distinction between public, criminal, and civil law [8].

In the FSCS, there is a distinction made between FSCD and BGE. FSCD are binding rulings made by the Swiss Federal Court in a specific case. These rulings address a particular legal issue and directly impact the parties involved. Meanwhile, BGE are general decisions made by the Federal Court judges to clarify legal questions. They do not affect specific disputes but serve as a guide for future court decisions and are binding on all Swiss courts. Unlike specific Federal Court decisions, leading decisions have wider significance and can impact the interpretation and application of laws in Switzerland.

2.5.1 Publications of Supreme Court Decisions

All BGE cases since 1954 have been published in anthologies and are organized by year and legal area. They can be accessed on the Swiss Supreme Court's webpage. The Swiss Supreme Court has published all FSCD cases since 2007, some cases from 2000-2007 are missing.[3]

Naming Conventions

Each case at the FSCS is assigned a unique file name with a specific syntax. The file name for FSCD and BGE cases consist of:

- FSCD: 9C 466/2021
  - A digit representing the responsible law department (9)

[2] If there are 500 relevant documents for a query, a maximum Recall score of 0.2 (R@100) would be produced even though all relevant documents were retrieved.

[3] https://www.bger.ch

- A cipher indicating the procedure (C)
- A consecutive number of up to four digits (466)
- The year of the case (2021)

- BGE: 148 V 385
  - A number representing the year, with 1 representing 1873 (148 represents 2021)
  - A Roman numeral indicating the volume (V)
  - A sequential number indicating the page number in the volume (385)

All BGE used to be a FSCD and can be found as BGE and FSCD. Their filenames are distinct and there is no connection between them making it difficult to find the corresponding FSCD for a given BGE file name or vice versa. Additionally these file names are used for referencing cases in subsequent court rulings.[4]

2.5.2 Sections

Every case at a Swiss court is divided into various text sections. For example, these sections may be separated by titles such as "Urteilskopf" for the header or "Regeste" for considerations. We differ between the following sections: **Header**: Introduction to the respective judgment and parties, containing name of the court, its composition, place and date. **Facts**: Present the disputed and undisputed facts and the request. **Considerations**: Court makes legal assessment of the entire events and legal reasoning which leads to the decision. **Rulings**: Contain the decision, merits and the costs. **Footing**: Contains information on legal remedies, sometimes also found at the end of rulings section.

[4] A comprehensive description of the naming conventions for Swiss Court Decisions can be found here: https://www.bger.ch/files/live/sites/bger/files/pdf/Divers

Chapter 3

Related Work

Legal Datasets

There is not yet that much Swiss legal data published. A dataset of Swiss Court Rulings was released by Niklaus et al. [28]. It includes 85k multilingual cases from the Federal Supreme Court of Switzerland. There is an overlap between datasets introduced in this thesis and Swiss-Judgment-Prediction, most of their cases are included and others added their work. Legal Data from the European Union is also available as the ECtHR dataset which includes 11k cases from the European Court of Human Rights was published by Chalkidis et al. [3]. This dataset is monolingual only containing English texts. In further work Chalkidis et al. [5] published the MULIT-EURLEX dataset containing 65k EU laws translated in 23 different languages. Among those 23 languages are English, German, French and Italian.

Text Classification

One of the most common text classification tasks in the legal domain is Legal Judgment Prediction. With Swiss data this was done by Niklaus et al. [28] using a binary label. They showed using hierarchical language model architecture to enable long input tokens is beneficial. Their experiments have shown that large multilingual language models like Roberta score the best results. Chalkidis et al. [2] introduced additionally to the Judgement Prediction the Importance Prediction task, which predicts the importance of a ECtHR case on a scale from 1 (key case) to 4 (unimportant). Legal experts defined and assigned these labels for each case, representing a significant contrast to our approach where labels were algorithmically determined. This is to our knowledge the only comparable task to criticality prediction.

Information Retrieval

A widely used IR technique is BM25, which is an improved retrieval method that considers the term frequencies and takes into account the saturation effect and document length. [33] For many years such term based models have scored SotA results. Thakur et al. [37] proposed a novel evaluation benchmark for IR that encompasses a wide range of approaches, including BM25, dense, and re-ranking models. They found that the BM25, although computationally expensive, provided a robust baseline, while other models did not achieve comparable performance. Their findings suggest that there is still much room for improvement in this area of NLP. Efficient retrieval of relevant information is crucial for many NLP tasks, and these results highlight the need for continued research in this area. [4] proposed a new IR task called REG-IR, which deals with longer documents in the corpus and entire documents as queries. This is an adaptation of doc2doc IR task, which aims to identify a relevant document for a given document. The authors observed that neural re-rankers underperformed due to contradicting supervision, where similar query-document pairs were labeled with opposite relevance. Additionally, they demonstrated for

long documents that using BM25 as a document retriever in a two-stage approach often results in underperformance since the parameters k and b are often not optimal when using standard values. The problem of noise filtering of long documents was also addressed by using techniques like stopwords removal. However, as seen in [21], this approach can have a negative effect on performance. The best pre-fetcher for long documents in Chalkidis et al. [4] was found to be C-BERTs, which are trained on classifying documents using predefined labels.

Chapter 4

Methods and Approach

To achieve the goal of this thesis, which is to propose two NLP tasks utilizing data from the Swiss Federal Supreme Court (FSCS), several steps are required. First, we need to download cases and store them in our local database (pipeline). This collected corpus serves as the basis for our tasks. Next, we need to process the data for each task in order to create the datasets that will be used for training and evaluation. This involves extracting the necessary information from the FSCD, such as the text, labels, and any other relevant features. We also need to consider any preprocessing steps that may be required, such as cleaning the text or performing language-specific tasks.[1] Once the datasets are prepared, they are uploaded to Huggingface. Further we need to adapt the two datasets to be used for either tasks. We intend to evaluate our tasks on various models and demonstrate the beneficial impact of additional pre-training of a model on legal language on the outcomes. The various steps involved in data processing are illustrated in Figure 4.1, which provides an overview of how the data is processed at each stage.

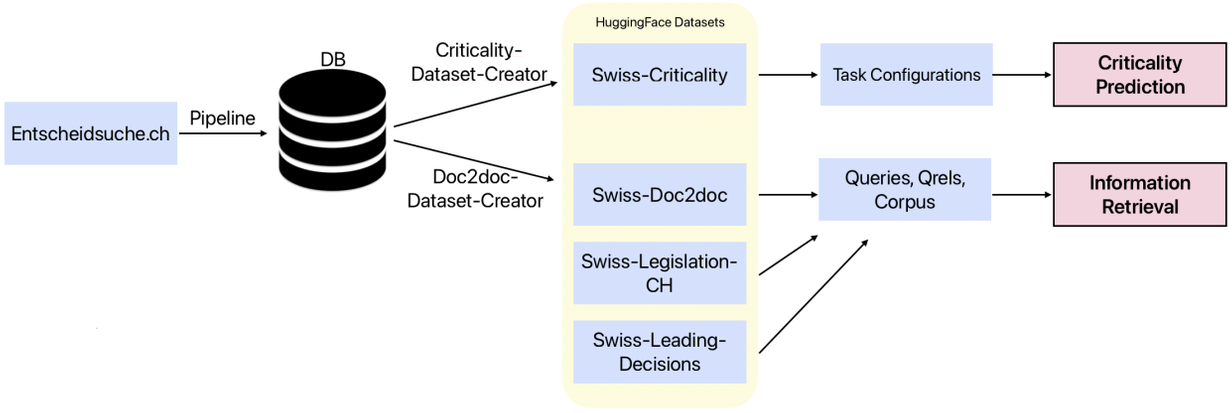

Figure 4.1. Workflow Overview

4.1 Pipeline

First, legal data was collected. Every day, new cases are published on Entscheidsuche.ch, enabling us to fetch new documents daily (Figure 4.2).

(1) We scraped all files found on Entscheidsuche.ch, which include metadata of every court's folder. Only case documents that do not already exist locally in our database are sent through the pipeline. We use BeautifulSoup's parsing and python's Requests library to extract the complete list of URLs to each of the case documents. The list is then iterated on to download all the files. **(2)** In a second step we create the required tables and save the extracted information. Again

[1] The code can be found on GitHub `https://github.com/JoelNiklaus/SwissCourtRulingCorpus`

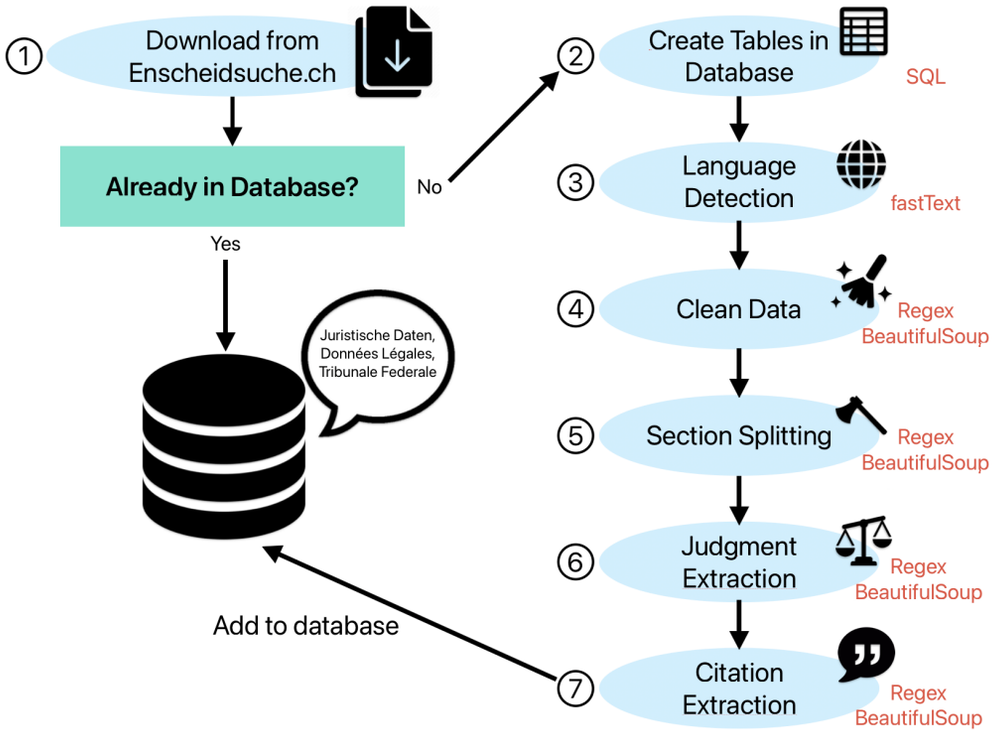

Figure 4.2. Pipleline of scraping Swiss Legal Documents

BeautifulSoup is used to extract the textual contents of HTMLs. For PDFs, we use the tika-python library to extract the document's content. **(3)** To extract the corresponding language, we use the fastText language identification tool [12]. Each case must have an appropriate language for further extraction tasks. **(4)** A cleaner is used to remove any strange patterns or redundant text to avoid errors for further extraction tasks. **(5)** Cases are split into the sections header, facts, considerations, rulings, and footer using a set of regex patterns. Each section has its own indicators that implicate the start of one section. **(6)** The judgment outcome is extracted, which can be one of *Approval*, *Partial approval*, *Dismissal*, *Partial Dismissal*, *Inadmissible*, *Write-Off*, and *Unification*. To map a ruling to one of the defined judgment outcomes, a set or combination of words is defined for each one of them. Since indicators for the different judgment outcomes are not exclusive to this context, it is crucial to consider only the ruling section of a case to avoid false positives as much as possible. Therefore, successful judgment outcome extraction is dependent on precise section splitting. **(7)** Citations are procured either via Regex (for cantonal cases) or BeautifulSoup (for federal cases). The FSCS labels all citations with an HTML tag, ensuring a high quality of citations for federal cases. A differentiation is made between law citations and Supreme Federal Court Decision (BGE) citations.

4.2 Criticality Prediction

The first task involves predicting the "criticality" of a FSCD through text classification. As there is no available expert labeling for criticality and it can be interpreted in various ways, we propose our own criticality labels. This approach not only focuses on identifying critical cases, but also differentiating between them, which is a challenging task even for legal experts. We introduce two approaches to set labels.

4.2.1 Criticality Dataset Creator

Criticality can be defined in numerous ways depending on the context. Ideally, legal experts should determine criticality based on predefined rules objectively, regardless of court, language,

or judge. However, for this project, labels are set algorithmically since expert labeling are unavailable. The goal is to establish a straightforward method for setting meaningful labels. To achieve this, the complexity of criticality was simplified by defining quantifiable rules for labeling. The following two approaches were initially considered but were ultimately dismissed for the reasons stated below:

- The Swiss Judgment Prediction Model [28] could be utilized, labeling cases as critical every time it is difficult to predict whether a case will be approved or dismissed. However, this idea was dismissed as it may be too easy for the model to learn how the Swiss Judgment Prediction Model classifies cases.
- FSCD originate in a lower court. Defining cases from lower courts ending up at the FSCS as "critical" and others as "non-critical" was an initial idea. Given the varied legal structures across cantons and law areas, where we have a different amount of instances before the FSCS, it is difficult to find a consistent definition of criticality across all cantons.

We identified two concepts that helped us accomplish our goal of establishing meaningful labels.

BGE-label

A binary label with two classes, "critical" and "non-critical," will be used. There exist FSCD decisions that are additionally published as "Leading decisions," also known as BGE, and are considered critical as they are published by experts. Those FSCD cases will be labeled as critical, which requires the accurate extraction of FSCD file names from the headers of BGE cases. All other cases will be labeled as "non-critical" as depicted in Figure 4.3 number 1.

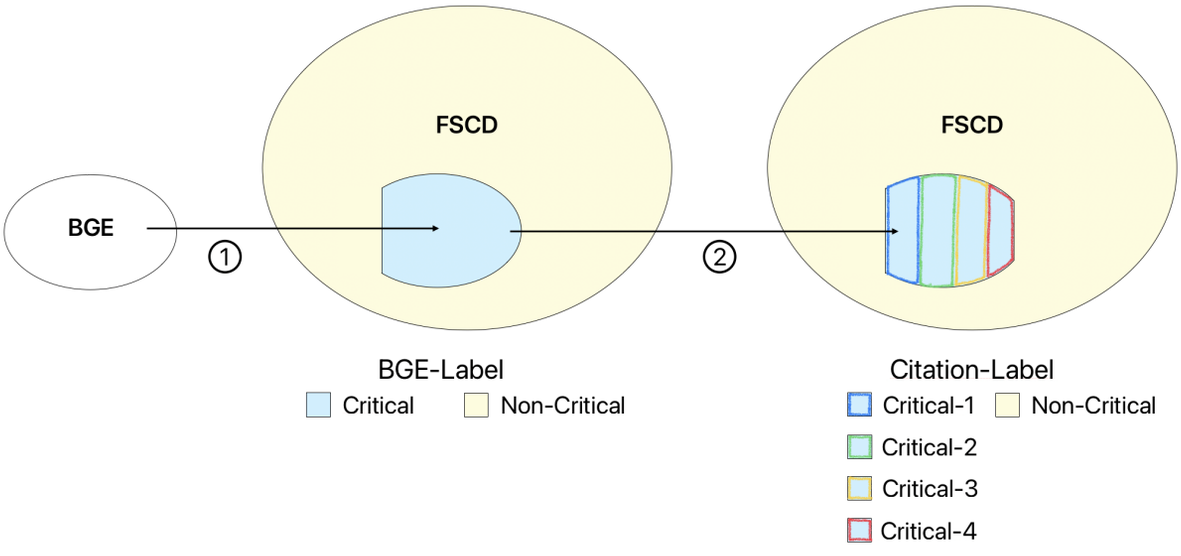

Figure 4.3. Relation of FSCD and BGE considering labels

Citation-label

The Citation-label (see Figure 4.3 number 2) is comprised of five different classes: "critical-1", "critical-2", "critical-3", "critical-4", and "non-critical." The "critical-1" class is considered the most critical, followed by "critical-2", "critical-3", and "critical-4" in descending order. BGE cases that are linked to and referenced in FSCD cases are deemed important as they impact more than just one case. We differentiate between old BGE cases and new cases, as there was less time for recent cases to be cited, which must be taken into account. To determine this

gradient of criticality, we went through each FSCD case and checked which BGE were cited in that FSCD. To avoid bias for each FSCD, we do not differentiate whether a BGE was cited once or 20 times, as this depends on the writing style of the author. In the next step, we aggregate those counts and get a score for each BGE. The counts were weighted based on the recency of the BGE case as follows:

$$weight = count * (year - 2002 + 1)/(2023 - 2002 + 1)$$

We separated the BGEs with the 25 percent highest scores into class "critical-1", those with the 25-50 percent highest score into class "critical-2", and so on. This approach resulted in a classification (visible in Figure 4.3 as blue, green, red, and yellow) with a roughly equal distribution of cases across the four critical classes. It is important to note that there may be cases with almost identical scores that are members of different classes. The combination of all four critical classes based on citation count is only a subset of cases that are considered critical according to BGE labels. This is because BGE cases which have never been cited were dismissed and labeled "non-critical".

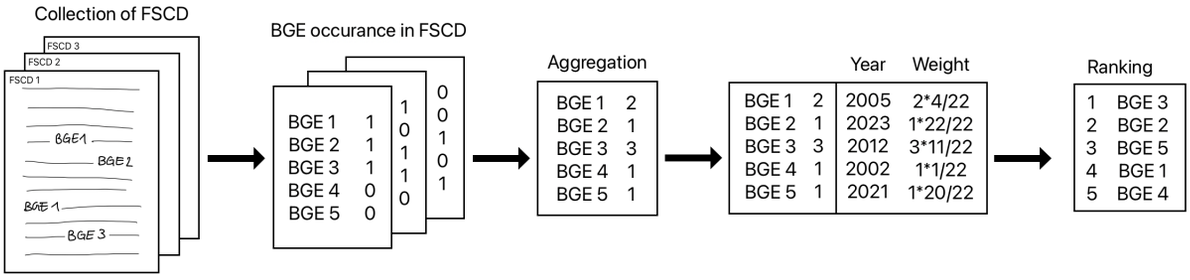

Figure 4.4. Method to create ranking of BGE

Splits

Before we can use our dataset to fine-tune a language model, we need to split it into train, validation, and test data. We split it in a stratified manner based on date rather than randomly. This approach was adopted to preserve the chronological order of the data, which is crucial for understanding the evolution of the law. The splits are as follows:

- Training: from 2002 to 2015
- Validation: from 2016 to 2017
- Test: from 2018 to 2022 (FSCD are only available until mid 2022)

The resulting dataset with its three splits is uploaded and published as Swiss-Criticality-Prediction dataset on HuggingFace. For more details on how the dataset was created, refer to SwissCourtRulingCorpus.

4.2.2 Task Configurations

Having our Swiss-Criticality-Prediction dataset we have to format data into the required format of input-target pairs. We developed four difference task configurations, where we define input and target differently. As input we are using either Facts or Considerations and provide BGE-label (critical or non-critical) or Citation-label (critical-1, critical-2, critical-3, and critical-4), with non-critical cases being disregarded for the Citation-label. We removed cases where the text-input is empty. Tasks are named BGE-Facts (BGE-F), BGE-Considerations (BGE-C), Citation-Facts (Cit-F) and Citation-Considerations (Cit-C).

4.2.3 Experiment Set Up

We are using each of our task configurations to fine-tune different language model and compare their results. Due to the long length of the input Facts or Considerations of a case, traditional models are not appropriate. Hence, we will use hierarchical models, as demonstrated by the superior results reported in [28] and [3]. Model hyperparameter are adapted for each task configurations accordingly. To address class imbalance, we will employ an oversampling technique for each task. The task at hand involves working with multilingual data from the legal domain, including cases in German, French, and Italian. To effectively evaluate this data, we explore both common multilingual models, such as "xlm-roberta-base", as well as models pre-trained on legal data, such as "joelito/legal-swiss-roberta-base", which are expected to perform better. SwissBERT is additionally pre-trained on general Swiss data. Further we evaluate our tasks on different model sizes (small, base, large), where big models should outperform smaller models. Find a detailed overview of all models used for evaluation in Table 4.1. We report the Macro-F1 for our experiments.

Model	Source	Params	Vocab	Specs	Corpus	# Langs
MiniLM	Wang et al. [41]	118M	250K	1M steps / BS 256	2.5TB CC100	100
DistilBERT	Sanh et al. [34]	135M	120K	BS up to 4000	Wikipedia	104
mDeBERTa-v3	He et al. [15, 14]	278M	128K	500K steps / BS 8192	2.5TB CC100	100
XLM-Roberta base	Conneau et al. [7]	278M	250K	1.5M steps / BS 8192	2.5TB CC100	100
XLM-Roberta large	Conneau et al. [7]	560M	250K	1.5M steps / BS 8192	2.5TB CC100	100
X-MOD base	Pfeiffer et al. [30]	852M	250K	1M steps / BS 2048	2.5TB CC100	81
SwissBERT (XLM vocab)	Vamvas et al. [39]	306M	250K	364k steps / BS 768	Swissdox	4
Legal-Swiss-Roberta-base	ours	184M	128K	1M steps / BS 512	CH Caselaw/Legislation	3
Legal-Swiss-Roberta-large	ours	435M	128K	500K steps / BS 512	CH Caselaw/Legislation	3
Legal-Swiss-Longformer-base	ours	208M	128K	50K steps / BS 512	CH Caselaw/Legislation	3

Table 4.1. Overview of the specifications of the language model

4.3 Information Retrieval

The goal of our IR task is finding relevant laws and BGE (they build together the corpus) based on the facts of a FSCD (which serves as query). In this task, we focus solely on the facts section as input because citations typically appear later in the considerations section. This is a challenging task for several reasons: (a) Legal language is more complex than generic language. (b) The presence of cases, laws, and leading decisions in German, French, and Italian makes this an IR task requiring multilingual information retrieval (MLIR). (c) We are using an entire document as model input, which is expected to cause existing models to underperform, as demonstrated by Chalkidis et al. [4]. Through experiments on elementary models, we aim to demonstrate the poor performance of existing models and emphasize the need for other approaches for this specific task. We use models from the existing BEIR benchmark study by Thakur et al. [37].

4.3.1 Doc2doc Dataset Creator

The Swiss-Doc2doc-IR dataset consists of all FSCD, which we annotated with a list of citations represented by a unique identifier for a BGE or law article. Citations are presented as strings in our database, created by the pipeline. For each BGE citation we are extracting the corresponding filename, and for law citations we extract the name of the law to find those in the Swiss-Leading-Decisions or Swiss-Legislation datasets. Since laws are available in all three languages, we provide laws in each language corresponding to a citation.

The created dataset which includes links from a FSCD to its cited laws and BGE is uploded and puplished on Huggignface as Swiss-Doc2doc-IR dataset. For more details on how the dataset was created refer SwissCourtRulingCorpus.

Figure 4.5. Process citations found in FSCD

4.3.2 Structure Data into Query, Corpus, and Qrel

To conduct experiments we structure data of Swiss-Doc2doc-IR into queries and qrels. First of all, we randomly split the data into a train and test set. Then we create queries and qrels for both splits. Queries consist only of a unique FSCD id and its Facts. The qrels comprise the information of citations. For every relevant law or BGE which was cited, there exists an entry in the qrel with value 1. The corpus is the collection of all federal laws and all BGE. The unique Law-id / BGE-id is used as key and we store the law-text / Facts + Considerations as text. The title is only additional information which consists of the sr-number / file-number.

```
corpus = {  Law_id: { title:  "SR_number", text:   "law text"},
            BGE_id: { title:  "File_number", text:   "Facts + Considerations"},
            ...
          }

queries = {FSCD_id: "Facts",  ...   }

qrels = { FSCD_id: {Law_id : 1, BGE_id : 1, ... }, ... }
```

Figure 4.6. Structure of corpus, queries and qrels

As the complexity of the task increases with the growing number of documents in the corpus, we anticipate a decline in performance as more data is incorporated. To assess the impact of dataset modifications on performance, we conducted an ablation study by making minor adjustments to the datasets. An efficient model should perform well with both the adapted and

original datasets. Efficiently handling multilingual challenges is crucial for dealing with Swiss legal documents.

4.3.3 Experiment Set Up

The following models were chosen to experiment on. All are part of the Thakur et al. [37] benchmark.

Term-based

We chose the **BM25** model using ElasticSearch.[2] Term-based approaches should scale well to our long documents. However, term based models in general lack the ability to process the context of texts and suffer from the problem of lexical gap. Moreover, they cannot compare documents written in different languages like neural approaches can. BM25 proved to provide a solid baseline for IR tasks.

Neural Approaches

We evaluate different multilingual **S-BERT** models. Next to experiments on existing S-BERT models as "distiluse-base-multilingual-cased-v1", we train our own S-BERT models, which is expected to better adapt to our task. In addition to normal training, we explore a training technique that involves using hard negatives [43]. During normal training, the model receives a query and a document and aims to optimize its weights to assign a high score if the document is relevant or a low score if it is irrelevant. For training with hard negatives, we first used another S-BERT model to generate "hard" negative examples. Those hard negative examples are documents that were mistakenly considered relevant by the other S-BERT model. The training process involves presenting the model with a query, a relevant document, and a hard negative document. This way the model can further improve its ability to distinguish between relevant and irrelevant documents, hopefully enhancing its performance. S-BERT are in general limited in their input length, we face the problem of our input texts being too long, which results in loss of context for our documents as they are truncated. [26] Another chosen approach is using a re-ranker model consisting of **BM25 and a Cross Encoder**, as they scored promising results for the BEIR benchmark. This model was dismissed for many experiments, as it is computationally expensive for a big corpus and long text inputs. [19] Further we chose to evaluate a **Dense Dimension Reduction** model using the S-BERT model "distiluse-base-multilingual-cased-v1". This model speeds up computation while still maintaining the advantages of a dense approach.

Metrics

Thakur et al. [37] used the NDCG to compare results across various tasks. However, we found that the Capped Recall@k (Rcap@k) score provides another great representation of the success of the models in our task. This is because each query has multiple relevant documents, and there is no need for ranking within those documents. The adjusted version of the Capped Recall@k score also accounts for the fact that each FSCD has a different number of citations.

Dataset Adaptions

The train set is only used to train S-BERT models. To evaluate models we used the test set or an adaption of it. Adaptations are, for example, the use of a subset of 100 queries (100), the

[2] https://www.elastic.co/elasticsearch/

use of queries in a specific language (DE/FR/IT), the removal of stopwords (S) or the use of monolingual links (SL) where the query and relevant documents in the corpus are written in the same language. An efficient model should be capable of performing well with both mono- and multilingual links, which is particularly crucial in Switzerland due to the numerous multilingual challenges we encounter.

Chapter 5

Datasets

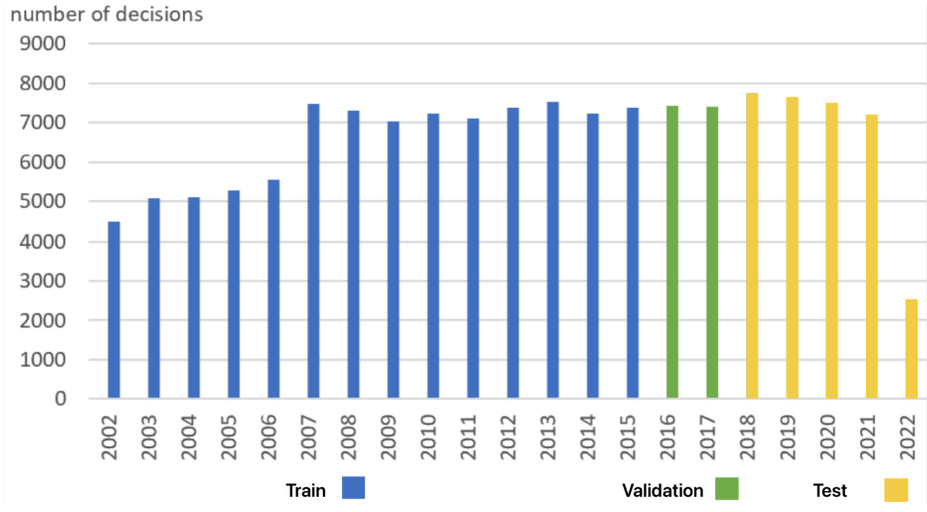

Figure 5.1. FSCD distribution over the years

In the following we present our two HuggingFace datasets Swiss-Criticality-Prediction and Swiss-Doc2doc-IR consisting of FSCD. Figure 5.1 shows that FSCD cases are distributed uniformly for most years - especially for 2007 until 2021. Additional we will use BGE cases from the existing datasets Swiss-Leading-Decisions and federal laws from the Swiss-Legislation-CH dataset (CH indicating using only federal laws). All datasets are publicly available on HuggingFace.[1] Cases were processed using the pipeline described in Section 4.1, the newest cases are from summer 2022. We can be certain to have all FSCD since 2007 and an incomplete collection of prior cases. Additional medatdata is collected as the five languages - German (DE), French (FR), Italian (IT), Romansh (RM), and English (EN). Additionally we reported the mean token length (MT) (see Chapter 2.1.1) of sections Facts and Considerations (see Chapter 2.5.2). As sections Facts (Fac) and Considerations (Con) are not available for Swiss-Legislation-CH due to other format, we decided to report the token length of the full text and marked it with *.

Dataset	Total	DE	FR	IT	RM	EN	Fac MT	Con MT
Swiss-Criticality	139K	85K	45K	8K	-	-	828	3K
Swiss-Doc2doc-IR	141K	87K	46K	8K	-	-	847	3K
Swiss-Leading-Decisions	21K	14K	6K	1K	-	-	689	3K
Swiss-Legislation-CH	16K	5K	5K	5K	207	132	-	7K*

Table 5.1. HuggingFace Datasets Overview

[1] https://huggingface.co/rcds

When comparing the section lengths of Facts and Considerations, it was found that Considerations are significantly longer than Facts, see the exact distributions in the Appendix B.1, B.2, B.3 and C.1. This is also observed with the median token length reported in Table 5.1. Given that considerations provide more context and text, we expect it to be easier to predict a class based on considerations rather than on facts.

5.1 Swiss-Criticality-Prediction Dataset

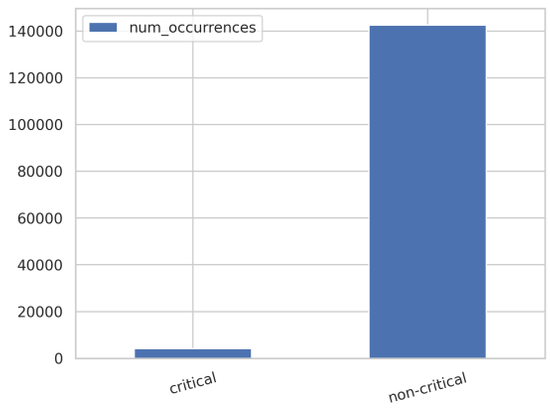

Figure 5.2. BGE-label distribution

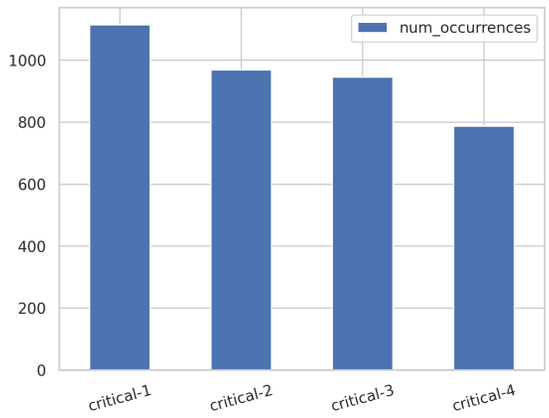

Figure 5.3. Citation-label distribution

Our dataset is heavily skewed towards "non-critical" cases for the BGE-label depicted in Figure 5.2. For the Citation-label only critical cases were considered resulting in a much smaller dataset with more uniformly distributed Citation-labels as depicted in Figure 5.3. Another imbalance can be seen in languages with roughly 60 percent of the cases being in German, while Italian cases are scarce reported in Table 5.1. Our dataset was split into train, validation (val) and test in a stratified manner. Further, we created four different dataset configurations to experiment on which are depicted in Table 5.2. Label names are *critical* (C), *non-critical* (NC), *critical-1* (C1), *critical-2* (C2), *critical-3* (C3), *critical-4* (C4).

Config	**Train**	**Labels Train**				**Val**	**Labels Val**				**Test**	**Labels Test**			
		C	**NC**				**C**	**NC**				**C**	**NC**		
BGE-Fac	**75K**	3K	72K	-	-	**12K**	580	13K	-	-	**26K**	950	25K	-	-
BGE-Con	**91K**	3K	85K	-	-	**15K**	580	13K	-	-	**32K**	948	29K	-	-
		C-1	**C-2**	**C-3**	**C-4**		**C-1**	**C-2**	**C-3**	**C-4**		**C-1**	**C-2**	**C-3**	**C-4**
Citation-Fac	**2.5K**	782	626	585	513	**563**	186	152	131	94	**725**	137	177	224	187
Citation-Con	**2.5K**	779	624	586	520	**563**	186	154	131	92	**723**	137	177	224	185

Table 5.2. Criticality Task Configurations

5.1.1 Sources of Errors

There are several sources of errors in this process. One is the potential to extract an incorrect filename from the BGE header. Additionally, finding the corresponding FSCD case for each filename is not always possible, those steps are depicted in Figure 5.4. Particularly for cases prior to 2007 cases ca not be found due to incomplete publication of older FSCD cases, this is reported in Figure 5.5. Since the file names are provided by Entscheidsuche.ch, we do not expect any errors there. As filenames are not unique, there may be multiple FSCD cases linked to the same BGE filename. If there was a FSCD being labeled "non-critical" while it actually is "critical", this would be a problem. However, the errors found, end in FSCD cases missing

and not being labeled incorrectly. Since this is not expected to have a negative impact on the experiments, we disregard those errors. Further we marked the different law areas to enhance that no law area was visible preferred in being labeled more critical than another.

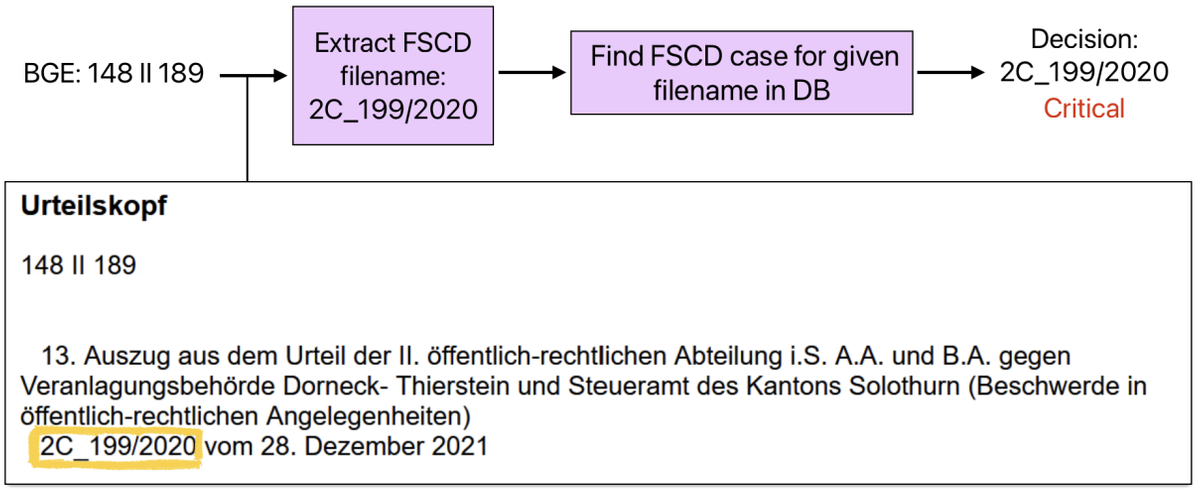

Figure 5.4. Different steps to find FSCD for BGE cases

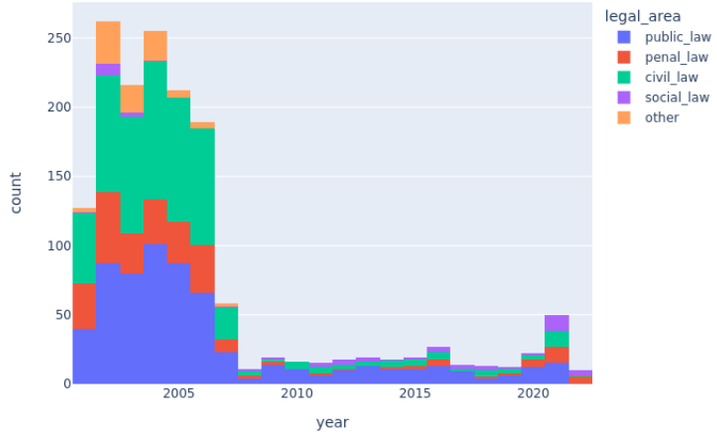

Figure 5.5. References found in BGE headers but not found as FSCD

In order to set the Citation-label we use the weighted BGE score. Detailed reports of the found score distributions are shown in Appendix Figure B.4 and B.5).

5.2 Swiss-Doc2doc-IR Dataset

For the Swiss-Doc2doc-IR dataset we are focusing on law and BGE citations in FSCD. Analyzing the amount of citations separated for laws and BGE shows, that there are quite big differences in the amount of citations (see Table 5.3 below). There could be found more BGE citations than laws, because there are laws missing in the Swiss-Legislation-CH dataset.

Further we structured our data into queries, qrels and corpus. The collection of all laws and BGE from the Swiss-Legislation-CH and Swiss-Leading-Decisions datasets serve as the corpus resulting in around 10K entries. We take the Facts of FSCD as a proxy for an appeal being written by the lawyer, they are used as queries. Cases in all three languages serve as queries. We

Split	Median	Mean $\pm Std(min - max)$
BGE	8.41	$8.67 \pm 6(1 - 175)$
Laws	3.66	$6.64 \pm 6(2 - 39)$

Table 5.3. Distribution of the number of laws and BGE citations in FSCD

excluded FSCD without valid citations. Qrels were created using citations of each FSCD, they could also be described as query-corpus pair. As laws are written in all three official languages, we end up with cross-lingual query-corpus pairs. One law citation will end up mostly in three query-corpus pairs, one for each language, while a BGE mostly exists only in one language. The mean token length of our queries mirror tasks like EU2UK by Chalkidis et al. [4], and is significantly longer than that in other IR benchmarks (like the SCIDOCS dataset of the Thakur et al. [37] benchmark (Table 5.4). This is primarily because we use entire documents as queries, which introduces challenges in our experiments.

Dataset	Queries length	Corpus length	Type
SCIDOCS	9	176	
EU2UK	1'849	2'642	
Swiss-Doc2doc-IR	847	689	BGE
		6'870	Laws

Table 5.4. Comparison of queries and corpus length distribution of different IR datasets

Chapter 6

Experiments

6.1 Criticality

This section presents the experiments and results for the Swiss-Criticality-Prediction task, which involves predicting a Single Label Classification Task (SLCT) using either the BGE-label or the Citation-label. We experimented on four distinct task configurations: BGE-Facts (**BGE-F**), BGE-Considerations (**BGE-C**), Citation-Facts (**CIT-F**), and Citation-Considerations (**CIT-C**). Tasks are evaluated on 10 multilingual language models.

6.1.1 Hyperparameters

In analyzing the input length for both BGE-label and Citation-label, we observed in Table 5.1 mean token lengths to be around 800 / 3200 for Facts and Considerations. Further the 75 percent quartiles were found to be 1'200 and 4'500 tokens. Using this information, we established the input parameters for Facts to allow 2048 (16x128) tokens and for Considerations to allow 4096 (32x128) tokens using a hierarchical BERT model. An example of the BGE-Consideration configuration parameters can be found in Figure 6.1.

```
"swiss_criticality_prediction_bge_considerations": {
  "DataTrainingArguments": {
    "max_seq_length": 512,
    "max_segments": 32,
    "max_seg_length": 128,
    "add_oversampling": true
  },
  "ModelArguments": {
    "hierarchical": true
  }
},
```

Figure 6.1. Example for parameters

For the experiments, we utilized Early Stopping mechanism with an initial learning rate of 3e-5. Additionally examples were over-sampled to avoid overfitting. The batch size was adjusted for each task, language model, and system. To ensure the validity of our results, we ran each experiment with three different random seeds. The results are presented in terms of the Macro F1 measure (where 1 is the best and 0 the worst score) and are shown in Table 6.1. All code used for experiments can be found on Github.

6.1.2 Results

Here are the reported Macro F1 scores for our Criticality Prediction task, distinguishing between general models and domain-specific pre-trained models. Additionally we report the Aggregated Score (Agg) across all tasks in the SCALE benchmark.

Model	Agg	BGE-C	BGE-F	Cit-C	Cit-F
MiniLM	32.4	54.7	65.8	9.8	20.8
DistilBERT	42.1	56.2	65.4	19.6	22.1
mDeBERTa-v3	40.2	55.1	69.8	21.0	17.5
XLM-RoBERTa-base	44.6	57.2	65.9	21.3	23.7
XLM-RoBERTa-large	**48.4**	56.4	67.9	24.4	**28.5**
X-MOD base	41.9	56.6	67.8	20.0	20.6
SwissBERT (XLM vocab)	44.6	56.2	67.3	**25.7**	23.0
Legal-Swiss-RoBERTa-base	46.9	57.6	**72.8**	23.1	22.5
Legal-Swiss-RoBERTa-large	46.2	57.4	70.8	21.3	23.3
Legal-Swiss-Longformer-base	42.8	**58.1**	70.8	21.4	17.4

Table 6.1. Criticality Prediction main results

6.1.3 Discussion

The pre-trained model, Legal-Swiss-RoBERTa-base, outperformed XLM-RoBERTa-base, suggesting that domain-specific pre-training can lead to significant improvements in model performance. Overall, pre-trained models showed better aggregated results compared to general models with the same size. However, contrary to expectations, the Legal-Swiss-RoBERTa-large model performed worse than its base counterpart. One plausible explanation for this discrepancy might be that the large model underwent merely half the pre-training steps compared to its base counterpart. This result possibly suggests that extensive pre-training is more important than model size, a result consistent with the findings of Liu et al. [25] and Touvron et al. [38]. Surprisingly, despite receiving additional training on lengthy data compared to the hierarchical Legal-Swiss-RoBERTa-base model, the Legal-Swiss-Longformer model fell short of surpassing its performance.

Analyzing the different tasks, we observed that using considerations as input proved to be easier than using facts. Moreover, in general, models struggled to predict the Citation-label accurately, with their predictions not being more reliable than random choices. However, we need to keep in mind how we evaluated this task. We currently do not distinguish between different types of wrong predictions.

6.2 Information Retrieval

6.2.1 Results

We present the results for our IR task, evaluating the performance using NDCG@k and adjusted Capped Recall@k metrics for k set to 1, 10 and 100. In the first part, we focus on a subset of 100 queries and consider only the documents in the corpus that are relevant to at least one of these 100 queries. In the second part, we evaluate the experiments on the entire dataset, employing various dataset adaptions. It is important to note that a comparison of the results between the two configurations may not be meaningful, as the 100-query task is a simplified scenario compared to using the entire dataset.

We used the following abbreviations for model specifications: **distil**use-base-multilingual-cased-v1, sbert-**legal-xlm**-roberta-base, sbert-**legal**-swiss-**rob**erta-base. For the BM25 model a language analyzer must be chosen, which is indicated with German, French and Italian. Dataset adaptions are indicated with: (S) stopword removal, (SL) using only single language links,

Model	Additional	*Rcap*@1 ↑	*Rcap*@10 ↑	*Rcap*@100 ↑	*NDCG*@1 ↑	*NDCG*@10 ↑	*NDCG*@100 ↑
Dim Reduction	distil	0.00	1.59	5.43	0.00	1.17	2.41
Cross Encoder	distil	5.94	8.04	14.20	2.97	1.84	7.35
BM25	'German'	9.90	8.41	15.19	9.90	9.14	11.58
Existing S-BERT	distil	8.08	11.83	43.35	8.08	10.55	21.56
Train S-BERT	distil	22.22	30.35	84.38	22.22	25.72	48.66
Train S-BERT	legal-xlm	32.32	32.34	81.77	32.32	30.89	49.11
Train S-BERT	legal-rob	**36.36**	**35.68**	76.03	**36.36**	**34.54**	49.90
Train HN S-BERT	distil	27.27	33.94	**86.81**	27.27	30.09	**52.03**

Table 6.2. Results IR: using a subsets of 100 queries

Method	Additional	Adaption	*Rcap*@1 ↑	*Rcap*@10 ↑	*Rcap*@100 ↑	*NDCG*@1 ↑	*NDCG*@10 ↑	*NDCG*@100 ↑
BM25			8.38	6.43	15.76	8.38	6.66	10.23
BM25		S	**10.64**	7.57	16.47	**10.64**	8.04	11.33
BM25		SL	7.91	**9.99**	**32.46**	9.13	**9.65**	**18.03**
BM25	'English'		8.38	6.43	15.76	8.38	6.66	10.23
BM25	'German'		8.69	6.54	15.99	8.69	6.82	10.43
BM25	'German'	S	10.88	7.65	16.79	10.88	8.14	11.53
BM25	'German'	SL	8.05	**9.94**	**32.63**	9.293	**9.70**	**18.17**
BM25	'French'		**11.37**	7.74	16.54	**11.37**	8.34	11.51
BM25	'Italian'		10.08	7.12	16.29	10.08	7.58	11.02
Dim Reduction	distil		0.71	0.62	2.42	1.64	1.40	2.95
S-BERT	distil		0.9	0.75	2.64	2.06	1.70	3.31
Train S-BERT	distil		4.40	3.92	12.64	10.11	8.76	16.16
Train S-BERT	distil	S	**4.69**	**4.14**	13.39	**10.77**	**9.27**	**17.05**
Train S-BERT	distil	SL	1.79	3.92	**14.17**	4.03	6.17	12.91
Train S-BERT	legal-xlm		2.77	2.58	10.17	6.36	5.66	12.03
Train S-BERT	legal-rob		3.97	3.47	12.28	9.12	7.76	15.16
Train HN S-BERT	distil		**3.97**	4.46	13.36	**9.12**	9.21	**16.87**
Train HN S-BERT	distil	S	3.76	**4.75**	12.80	8.64	**9.66**	16.57
Train HN S-BERT	distil	SL	2.34	4.37	**14.43**	5.27	6.99	13.75
Train S-BERT	distil	DE	**4.22**	4.49	15.21	**8.21**	**8.15**	**15.86**
Train S-BERT	distil	DE SL	4.06	**8.47**	**29.43**	4.51	6.73	13.78
Train S-BERT	distil	FR	1.88	2.2	9.19	5.77	6.22	13.94
Train S-BERT	distil	FR SL	2.69	5.68	27.28	3.00	4.59	11.11
Train S-BERT	distil	IT	0.22	0.24	0.79	5.43	5.74	11.44
Train S-BERT	distil	IT SL	1.71	4.54	16.24	1.91	3.38	6.83

Table 6.3. Results IR: using dataset adaptions

(DE/FR/IT) using only queries in one language. The scores with optimal performance are highlighted per section. Training SBERT models using hard negative examples is indicated by (HN).

6.2.2 Discussion

We aim for a Capped Recall@100 score close to 100, indicating that our models can retrieve all relevant documents (on average around 20) within the first 100 retrieved documents. In comparison, NDCG scores also focus on the ranking of the first k documents, which is not as significant to our evaluation. While @1 scores provide some insight, they do not provide substantial information about the performance of our models. Therefore, our primary focus lies on Capped Recall scores at @10 and @100, as they provide more meaningful assessments of our models' ability to retrieve relevant documents.

General In general results reveal a consistent inability to retrieve the majority of relevant documents even when we set @k to 100. BM25 served as a robust baseline, even though Lucene Parameters were not optimized, suggesting the potential for even better results with parameter optimization [4]. Furthermore, we can observe that Dense Dimension Reduction was clearly outperformed by S-BERT and BM25 models. For S-BERT, Cross Encoders, and Dim-Reduction models, input truncation led to context loss, negatively affecting scores – an issue not encountered with lexical models. Training S-BERT models using Multiple Negative Ranking Loss [16] yielded significant performance improvement, with the use of hard negative examples proving advantageous.

Cross-Encoder Looking at Table 6.2, we can observe that the Capped Recall scores for BM25 and Cross Encoder are identical. This is expected since BM25 serves as a pre-fetcher in the retrieval process. Interestingly, all NDCG scores worsen when Cross Encoders are used as an additional re-ranker, indicating underperformance in this scenario. A reason why Cross Encoders fell short compared to BM25 could be due to inconsistent supervision, particularly when dealing with long legal texts, as highlighted by [4]. This is the case when sentences belonging to relevant and non-relevant documents are very similar.

BM25 When using BM25, it is possible to select an additional language analizer such as German, French or Italian language analyzer. Based on the chosen language, the language analizer incorporates specific stopword removal and word stemming techniques. Surprisingly, the French language analyzer excelled in multilingual lexical retrieval, despite the prevalence of German in our dataset.

Dataset Adaptions The evaluation of S-BERT on individual languages (DE, FR, and IT) highlighted its inconsistency in performance across different languages. This could be caused by the training set consisting of more German than French or Italian documents. Further experiments conducted on data containing only single language links (SL) demonstrated improved Capped Recall scores for all models, particularly when focusing on larger values of k. This indicates a general difficulty for the models in handling cross-lingual document links. Surprisingly, the removal of stopwords resulted in a performance boost for S-BERT models, which was unexpected. This can be explained by the limited input length, where the removal of stopwords allows the remaining words to capture more meaningful information instead of being overshadowed by common words. Additional experiments on data containing only single language links (SL) demonstrated improved Capped Recall scores for all models, indicating a general difficulty for models when handling cross-lingual document links. Stopword removal surprisingly revealed performance boost for S-BERT models, which was not expected. An explanation might be that the removal of stopwords allows the remaining words to capture more meaningful information. Despite the language analyzer for BM25 already including stopword removal, additional stopword removal improved performance. This suggests that the default stopword removal configuration of the language analyzer may not be optimized for our specific task, and additional customization is necessary.

Computation Speed An examination of evaluation speed revealed a significant advantage for the dimension reduction model, which completed its evaluation in approximately 10 seconds. Moreover, the S-BERT model, trained on distiluse-base-multilingual-cased-v1, took around 90 sec, exhibited a speed approximately four times faster than S-BERT based on other models.

Overall, our study exposes limitations of models in dealing with multilingualism, long documents, and legal texts, areas relatively underexplored in previous research. These findings provide a basis for the IR community to innovate strategies for these challenges.

Chapter 7

Conclusion

We have introduced two challenging multilingual tasks that hold significant potential and offer intriguing use cases. There is a pressing need for more advanced models capable of effectively handling longer inputs, multilingualism and evaluating context more efficiently.

Further Work To enhance model performance, we suggest leveraging libraries like Ray Tune for hyperparameter tuning. Additionally, we intend to investigate the creation of a regression task specifically for citation prediction. By doing so, we would assign varying penalties to different types of incorrect predictions, e.g. we distinguish, whether a very critical case, such as "critical-1", is wrongly predicted as "critical-4" compared to a scenario in which it is predicted as "critical-2". By utilizing this approach, we can more accurately evaluate the quality of predictions and gain deeper insights into the models' performance in this critical aspect. Further it would be interesting to explore IR models which are able to handle longer documents.

Ethical Concerns In addition to technical considerations, there are ethical concerns surrounding the rapid development of artificial systems in the NLP field. As these systems continue to revolutionize our world at an unprecedented pace, it is crucial to consider their potential impact on our future.

Appendix A

SCALE

This Bachelor thesis is part of the SCALE Benchmark by Rasiah et al. [31]. Our tasks Criticality Prediction and Information Retrieval are among 8 different tasks in scale as seen in Figure A.1

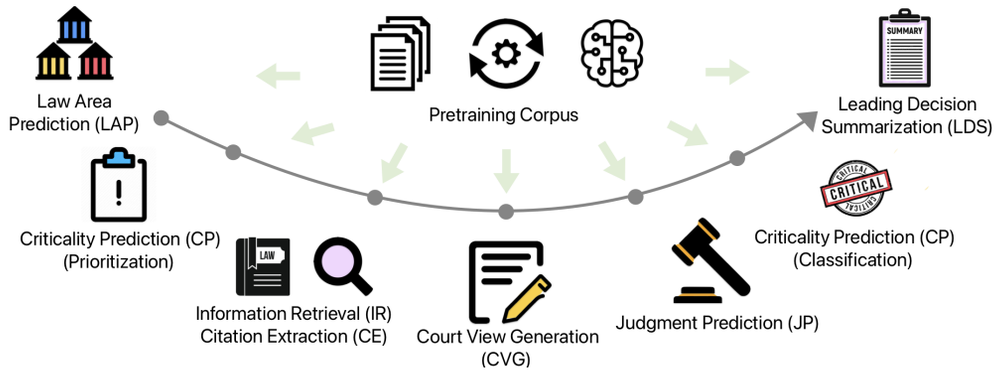

Figure A.1. SCALE

Appendix B

Dataset Distributions Swiss-Criticality-Prediction

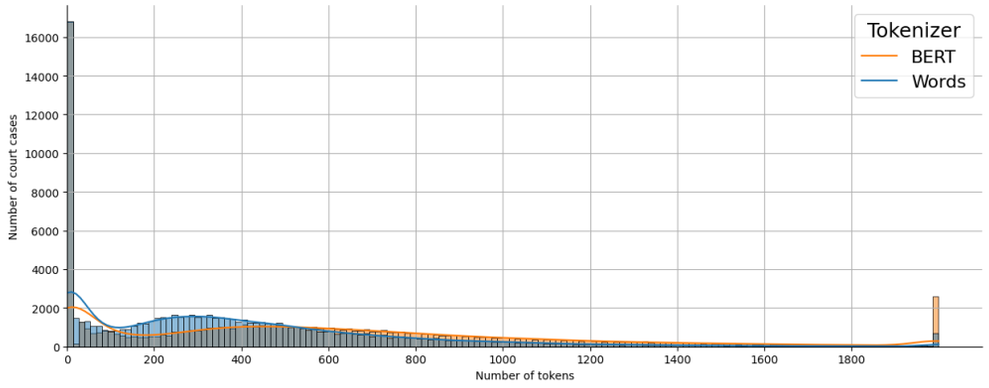

Figure B.1. Section facts input length distribution

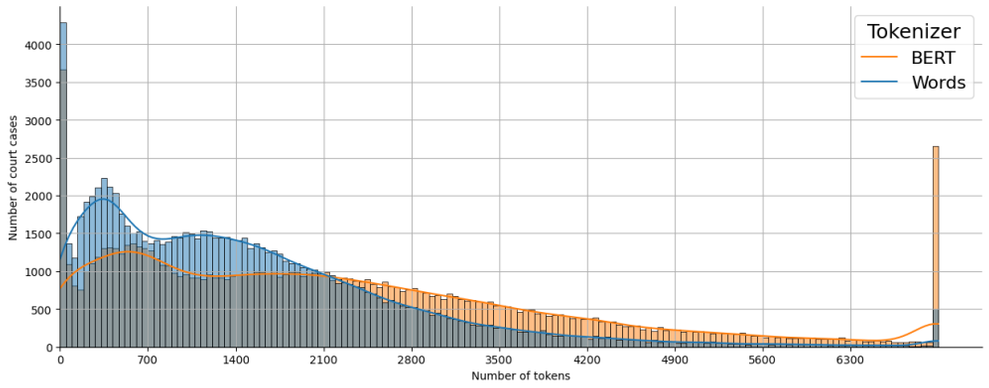

Figure B.2. Section considerations input length distribution

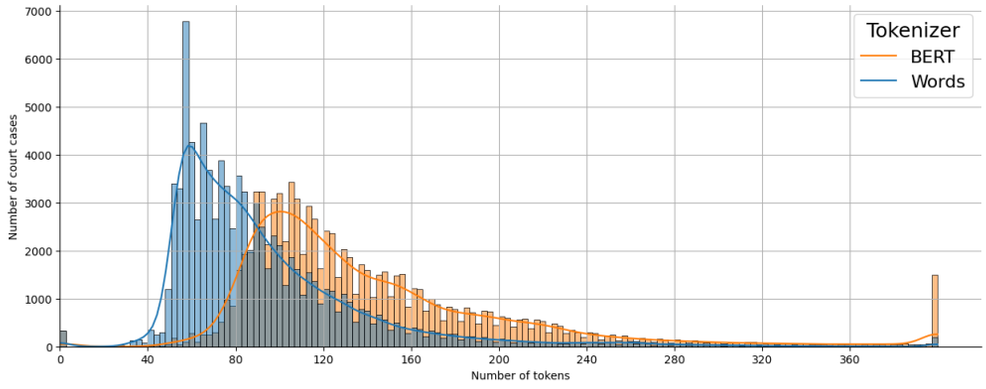

Figure B.3. Section rulings input length distribution

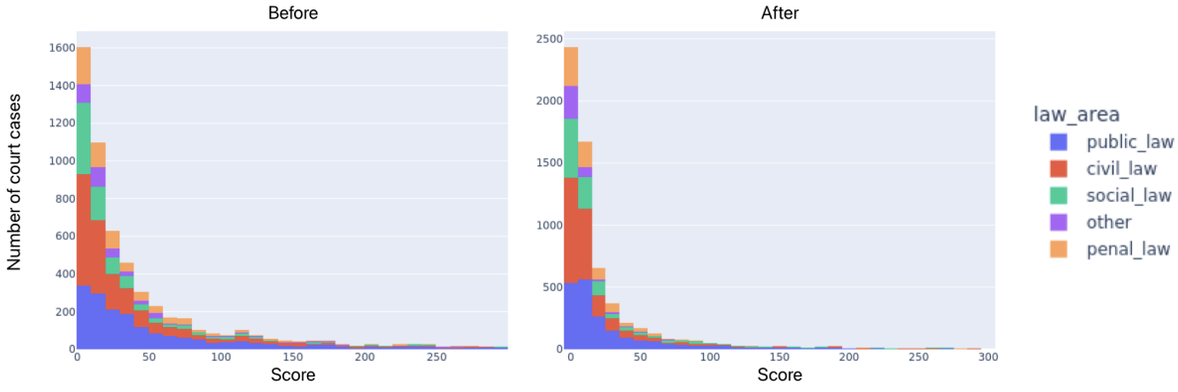

Figure B.4. BGE citation scores before and after weighting

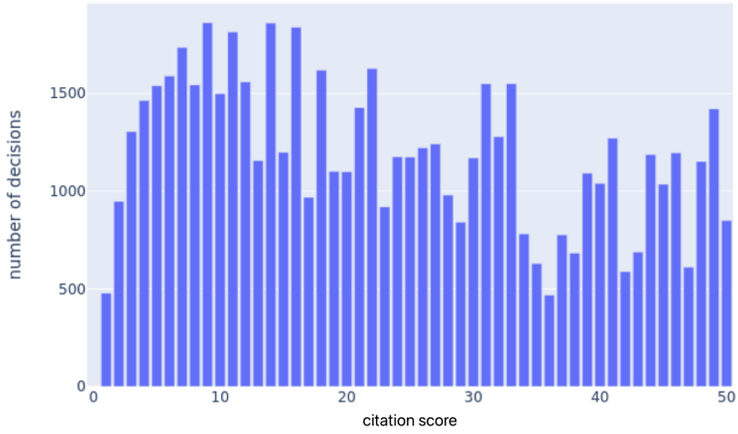

Figure B.5. Zooming into weighted BGE citation scores

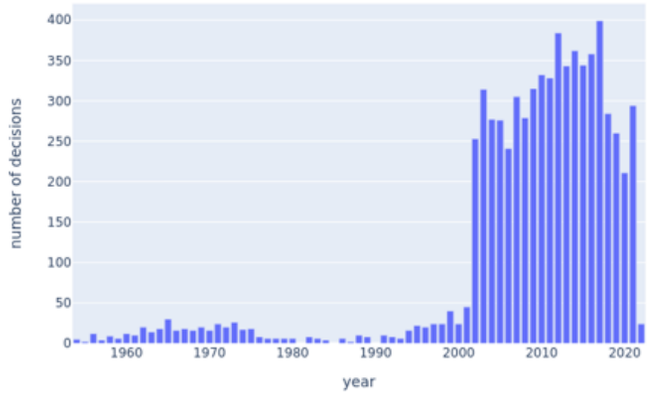

Figure B.6. Extracted references in BGE header per year

Appendix C

Dataset Distributions Swiss-Doc2doc-IR

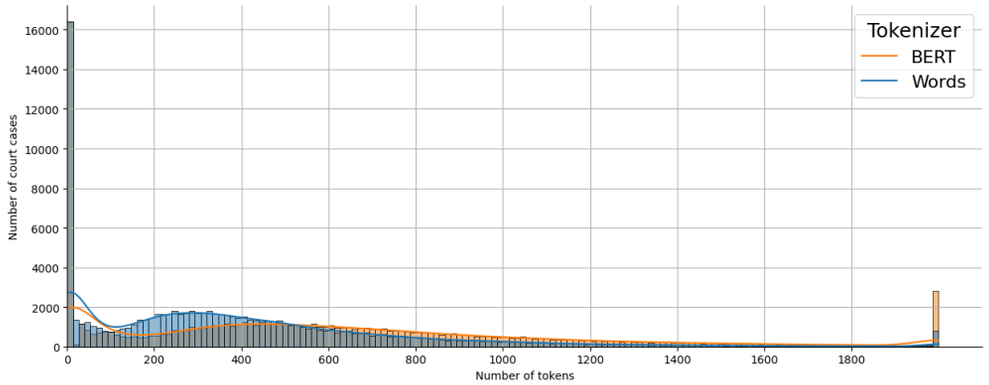

Figure C.1. Section facts input length distribution

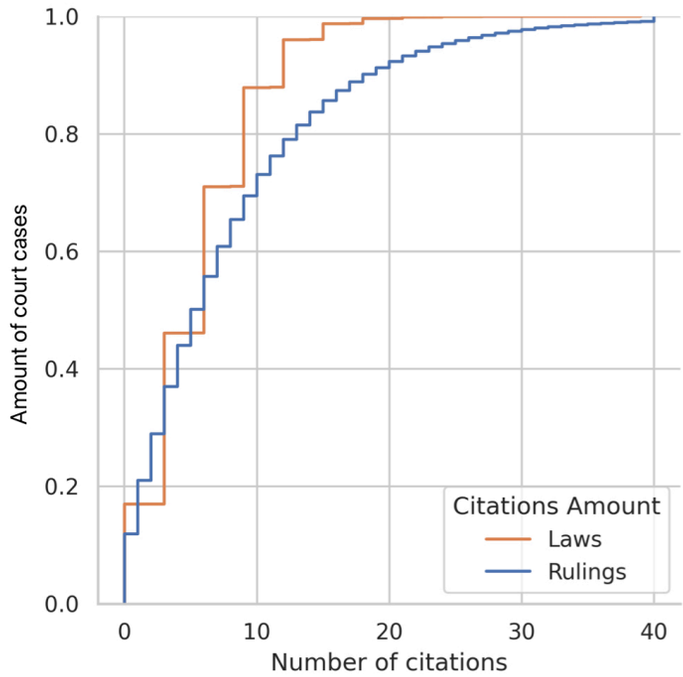

Figure C.2. Laws and BGE citation amount distribution

Bibliography

[1] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. Bridging the lexical chasm: Statistical approaches to answer-finding. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 192–199, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132263. doi: 10.1145/345508.345576. URL https://doi.org/10.1145/345508.345576.

[2] I. Chalkidis, I. Androutsopoulos, and N. Aletras. Neural legal judgment prediction in english. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, 2019.

[3] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, and I. Androutsopoulos. Neural contract element extraction revisited: Letters from sesame street, 2021.

[4] I. Chalkidis, M. Fergadiotis, N. Manginas, E. Katakalou, and P. Malakasiotis. Regulatory compliance through doc2doc information retrieval: A case study in eu/uk legislation where text similarity has limitations, 2021.

[5] I. Chalkidis, A. Jana, D. Hartung, M. Bommarito, I. Androutsopoulos, D. M. Katz, and N. Aletras. Lexglue: A benchmark dataset for legal language understanding in english, 2022.

[6] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019. URL http://arxiv.org/abs/1911.02116.

[7] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale, 2020.

[8] S. F. S. Court. Die wege zum bundesgericht, 2023. URL https://www.bger.ch/index/federal/federal-inherit-template/federal-rechtspflege.html. Accessed: 2023-03-13.

[9] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[11] P. Galuščáková, D. W. Oard, and S. Nair. Cross-language information retrieval, 2022.

[12] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[13] A. J. Hartig and X. Lu. Plain english and legal writing: Comparing expert and novice writers. *English for Specific Purposes*, 33:87–96, 2014. ISSN 0889-4906. doi: https://doi.org/10.1016/j.esp.2013.09.001. URL https://www.sciencedirect.com/science/article/pii/S0889490613000641. Special Issue : ESP in Asia.

[14] P. He, J. Gao, and W. Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. *arXiv:2111.09543 [cs]*, Dec. 2021. URL http://arxiv.org/abs/2111.09543. arXiv: 2111.09543.

[15] P. He, X. Liu, J. Gao, and W. Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *arXiv:2006.03654 [cs]*, Oct. 2021. URL http://arxiv.org/abs/2006.03654. arXiv: 2006.03654.

[16] M. Henderson, R. Al-Rfou, B. Strope, Y.-H. Sung, L. Lukács, R. Guo, S. Kumar, B. Miklos, and R. Kurzweil. Efficient natural language response suggestion for smart reply. *ArXiv*, abs/1705.00652, 2017.

[17] D. Hendrycks, C. Burns, A. Chen, and S. Ball. Cuad: An expert-annotated nlp dataset for legal contract review, 2021.

[18] Huggingface. Conceptual guides, 2023. URL https://huggingface.co/docs/transformers/tokenizer_summary.

[19] S. Humeau, K. Shuster, M.-A. Lachaux, and J. Weston. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring, 2020.

[20] D. M. Katz, D. Hartung, L. Gerlach, A. Jana, and M. J. B. I. au2. Natural language processing in the legal domain, 2023.

[21] J. Leveling. On the effect of stopword removal for sms-based faq retrieval. In *International Conference on Applications of Natural Language to Data Bases*, 2012.

[22] Q. Li and Q. Zhang. Court opinion generation from case fact description with legal basis. In *AAAI Conference on Artificial Intelligence*, 2021.

[23] Y. L. Li Deng. *Deep Learning in Natural Language Processing*. Springer Singapore, Springer Nature Singapore Pte Ltd. 2018, 2018. doi: https://doi.org/10.1007/978-981-10-5209-5.

[24] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

[25] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[26] P. M. Mekontchou, A. Fotsoh, B. Batchakui, and E. Ella. Information retrieval in long documents: Word clustering approach for improving semantics, 2023.

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[28] J. Niklaus, I. Chalkidis, and M. Stürmer. Swiss-Judgment-Prediction: A Multilingual Legal Judgment Prediction Benchmark. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 19–35, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.nllp-1.3.

[29] J. Niklaus, V. Matoshi, P. Rani, A. Galassi, M. Stürmer, and I. Chalkidis. Lextreme: A multi-lingual and multi-task benchmark for the legal domain, 2023.

[30] J. Pfeiffer, N. Goyal, X. V. Lin, X. Li, J. Cross, S. Riedel, and M. Artetxe. Lifting the Curse of Multilinguality by Pre-training Modular Transformers, May 2022. URL http://arxiv.org/abs/2205.06266. arXiv:2205.06266 [cs].

[31] V. Rasiah, R. Stern, V. Matoshi, M. Stürmer, I. Chalkidis, D. E. Ho, and J. Niklaus. Scale: Scaling up the complexity for advanced language model evaluation, 2023.

[32] N. Reimers and I. Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv:1908.10084 [cs]*, Aug. 2019. URL http://arxiv.org/abs/1908.10084. arXiv: 1908.10084.

[33] S. Robertson and H. Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. ISSN 1554-0669. doi: 10.1561/1500000019. URL http://dx.doi.org/10.1561/1500000019.

[34] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108 [cs]*, Feb. 2020. URL http://arxiv.org/abs/1910.01108. arXiv: 1910.01108.

[35] Subhadip Paul. Information retrieval with document re-ranking with bert and bm25, 2023. URL https://medium.com/@papai143/information-retrieval-with-document-re-ranking-with-bert-and-bm25-7c29d738df73.

[36] C. Sun, X. Qiu, Y. Xu, and X. Huang. How to fine-tune bert for text classification?, 2020.

[37] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models, 2021.

[38] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023. URL http://arxiv.org/abs/2302.13971. cite arxiv:2302.13971.

[39] J. Vamvas, J. Graën, and R. Sennrich. Swissbert: The multilingual language model for switzerland. *arXiv e-prints*, pages arXiv–2303, 2023.

[40] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017.

[41] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[42] D. Xu, I. E. H. Yen, J. Zhao, and Z. Xiao. Rethinking network pruning – under the pre-train and fine-tune paradigm, 2022.

[43] J. Zhan, J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1503–1512, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3462880. URL https://doi.org/10.1145/3404835.3462880.

Erklärung

Erklärung gemäss Art. 30 RSL Phil.-nat. 18

Ich erkläre hiermit, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Mir ist bekannt, dass andernfalls der Senat gemäss Artikel 36 Absatz 1 Buchstabe r des Gesetzes vom 5. September 1996 über die Universität zum Entzug des auf Grund dieser Arbeit verliehenen Titels berechtigt ist.

Für die Zwecke der Begutachtung und der Überprüfung der Einhaltung der Selbständigkeitserklärung bzw. der Reglemente betreffend Plagiate erteile ich der Universität Bern das Recht, die dazu erforderlichen Personendaten zu bearbeiten und Nutzungshandlungen vorzunehmen, insbesondere die schriftliche Arbeit zu vervielfältigen und dauerhaft in einer Datenbank zu speichern sowie diese zur Überprüfung von Arbeiten Dritter zu verwenden oder hierzu zur Verfügung zu stellen.

Bern 11.7.2023
Ort/Datum


Unterschrift