LexKey: A Keyword Generator for Legal Documents

Benjamin Cérat[1], Olivier Salaün[2], Noreddine Ben Jillali[1], Marc-André Morissette[1], Isabela Pocovnicu[1], Emma Elliott[1] and François Harvey[1]

[1]Lexum Inc., Canada
[2]RALI, DIRO, Université de Montréal, Canada

Abstract

Recent advances in natural language processing like large pre-trained transformer models have opened up a host of previously out-of-reach automation even to small businesses. The LexKey project is one such initiative, assembling a large dataset of annotated decisions from various sources and training an abstractive generative model that produces useful and well-formatted keywords from legal texts. We present the challenges involved and the steps taken to achieve this goal, from data cleaning to modifying an existing model architecture to handle long legal documents to the evaluation, both quantitative and qualitative, of the output.

Keywords

legal dataset, generative neural networks, keywords generation, long documents

Figure 1: Search results with keywords on CanLII [3].

1. Introduction

Recent years have seen the advent of transformer-based language models [1, 2] that have been applied to a wide variety of tasks across the field of natural language processing (NLP). The legal domain has not been exempt from this trend: automatic legal document classification, information retrieval and even summarization tasks have all become attainable goals. This paper outlines the step-by-step efforts required to accomplish one of these tasks, namely keyword generation, in a commercial setting.

Lexum is a small software company owned by the Canadian Legal Information Institute and focused on providing open access to online legal information through the canlii.org website and other related products.

Our objective was to generate, for each decision, short instructive keywords complying with the style found in Canadian legal reports [4] (some examples are shown in Table 1). Such keywords must focus on the legal questions raised in the decisions and not on factual details. They are then added inline to search results (light grey text in Figure 1) to provide users with more context on the content of the decision.

The LexKey project started life as a fairly basic proof of concept. The original prototype used a pre-trained model stored in Huggingface [5], namely BigBirdPegasus [6] (BBP) trained on the BIGPATENT dataset [7][1], which we fine-tuned to generate keywords harvested from decisions found in the Ontario Reports, the Law Society of Saskatchewan Libraries databases and the Supreme Court of Canada Reports. The ability to handle fairly long documents was paramount, as Canadian decisions can vary from several sentences to novel length, averaging around 6 thousand words.

While the preliminary results from BBP were promising, we outlined several issues regarding generated keywords. First, pre-trained language models capable of handling long documents were not available for the French language (one of the official languages of Canada), and the rare French legal models were either not accessible [8] or not suited for Canadian common law [9]. Second, the model fails to generalize to decisions from unseen courts. Inference on decisions from courts unseen in the training set was strongly biased toward topics found in reports (books that collect and publish notable case law), thus confusing jurisdictions and generally being of much lower quality. Finally, the keyword format is quite inconsistent across collections (e.g. length and styles differences among examples in Table 1) and the model generates keywords in a style that matches only one of those formats. Since the output is meant to be displayed inline on search results, a more uniform result is desir-

[1]Model checkpoint available at https://huggingface.co/google/bigbird-pegasus-large-bigpatent

Proceedings of the Sixth Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2023), June 23, 2023, Braga, Portugal.
ceratb@lexum.com (B. Cérat); salaunol@iro.umontreal.ca (O. Salaün); jillalin@lexum.com (N. B. Jillali); morissm@lexum.com (M. Morissette); pocovnicui@lexum.com (I. Pocovnicu); elliotte@lexum.com (E. Elliott); harveyf@lexum.com (F. Harvey)

Table 1
Example of Keywords from Different Sources

Decision source	Keywords
Supreme Court of Canada Reports	Constitutional law – Charter of Rights – Life, liberty and security of the person – Fundamental justice – Abortion – Criminal Code prohibiting abortion except where life or health of woman endangered – Whether or not abortion provisions infringe right to life, liberty and security of the person – If so, whether or not such infringement in accord with fundamental justice – Whether or not impugned legislation reasonable and demonstrably justified in a free and democratic society – Canadian Charter of Rights and Freedoms, ss. 1, 7 – Criminal Code, R.S.C. 1970, c. C-34, s. 251.
Law Society of Saskatchewan Libraries Databases	Criminal Law - Sentence - Robbery and Extortion
Canadian Federal Courts Reports	Citizenship and Immigration — Status in Canada — Convention Refugees and Persons in Need of Protection — Immigration practice — Refugee Appeal Division jurisdiction — Judicial review of Immigration and Refugee Board, Refugee Appeal Division (RAD) decision dismissing applicants' appeal from Refugee Protection Division (RPD) decision refusing to recognize applicants' claim to being refugees or persons in need of protection within meaning of Immigration and Refugee Protection Act, ss. 96, 97

able.

The LexKey project was meant to address these issues with the following contributions: first, we further pre-trained a multilingual model [10, 11] using a denoising task [12] on CanLII's Canadian legal decisions, doctrines and legislations. We then modified the model's attention to allow longer inputs and then fine-tuned it on a supervised keyword generation task over a large, carefully curated and normalized set of decisions containing keywords.

The model has been deployed live on CanLII in February 2023 after a lengthy quality analysis consisting of both automatic and manual evaluations performed by experts. The underlying pre-trained language model is also shaping up to be a crucial part of other related projects.

2. Related Works

2.1. Extractive Methods

Automatically generating structured keywords from textual documents is a task that is closely related to summarizing documents, a common task in natural language processing. Extractive statistical systems have long been the standard approach to perform this task. In fact, LexKey is aiming at replacing an in-house modified TF-IDF extraction system [13] that works by ranking sentences or n-grams in the text and picking a few salient expressions to represent the document. This extractive approach is similar to those used by [14, 15, 16].

Doing so prevents the model from inventing falsehoods about the document's content by constraining its output to segments from the input text. The traditional TF-IDF approach compares n-gram frequencies inside the document to those of the corpus at large. This selects groups of words that are rarely used but common in the document text. In our experience, our existing TF-IDF-based system applied to legal decisions yields keywords that are generally related to the facts of a case, but not to the legal principles discussed.

However, human-written summaries and keywords found in law reports (books that collect and publish notable case law) and legal databases, put an emphasis on broader legal doctrines, the pertinent case law and the statutes considered. The LexKey project aims at highlighting these concepts with new keywords.

There have been some efforts to leverage transformer-based models such as BERT to extract the most relevant sentences as a summary [17], but they struggle to perform as well as abstractive models of similar complexity.

2.2. Abstractive Methods

Since the advent of BERT [2], abstractive models using the transformer architecture [1] dominate the landscape of summarization tasks. Overall, most of the best performing models such as T5 [18], BART [12], MBART [10] and Pegasus [19] use an encoder-decoder architecture.

Several elements drew us towards encoder-decoder models: the flexibility provided by separately designed encoder and decoder layers, the flexibility in the implementation of the pre-training objective (e.g. masked language modelling [2], denoising [12]) and of the attention architecture [1, 6, 20], along with the ease of managing multilingual models by simply using source and target

language prompts [10].

More recently, at the time the LexKey project was already nearing its completion, large decoder-based models such as GPT [21] have also been shown to perform well in text generation tasks based on prompting schemes. The large-scale application of the most recently released models (i.e. ChatGPT [22], GPT-4 [23]) to our corpus is left for future work.

All in all, most of the architectures we surveyed were performing very similarly on summarizing tasks, so the ease of adapting the model to our needs (training and inference cost over millions of documents in particular) became the primary differentiator.

2.3. Handling Long Documents

Early transformer models like BERT [2] were limited to fairly short sequences (512 tokens) due to their quadratic memory usage as a function of the input length. Research into more efficient encoder attention mechanisms or implementation is very active: Flash Attention [24], Big Bird [6], Longformer-Encoder-Decoder [25], LSG [20] and LongT5 [26] are all fairly recent models implementing such techniques. Most of them allow extending input length up to around 16k tokens by making the memory usage scale more linearly in relation to the input length.

Hierarchical Attention architectures [27] have also been tried but seem to be limited to around 4k tokens inputs and require much more extensive architectural changes relative to the basic transformer implementation.

In the context of Multi-LexSum, a summarization task of civil rights lawsuits, [28] also emphasized the long-range input issues faced by transformer models when dealing with a multi-document case.

3. Datasets

3.1. Sources

One of the most important objectives of the LexKey project was to collate a representative set of annotated decisions that was large enough to train a highly capable keyword generation model whose output would meet the quality expectations of our users. Annotating by hand tens of thousands of decisions was unfeasible, so we decided to gather case law databases that already had keywords and categories added by experts.

We had a few criteria for the selection of appropriate training material. The most obvious one was that the legal framework that created them should be fairly similar to Canadian Law. This meant limiting ourselves to countries that are under common law (mostly the Commonwealth member countries and to some degree the United States). The decisions also had to be either in French or English and have a keyword format somewhat similar to the style we were aiming at in order to ease format normalization.

Gathering decisions from Law Reports introduced a fairly strong bias toward decisions that were of interest to legal practitioners, which feature appeal cases, decisions that are considered authorities on broad constitutional questions, or decisions that clarify a legal controversy. These differ from day-to-day decisions without precedential value.

Table 2
Decisions with Keywords per Source

Source	Count	%
Ontario Reports	20 463	13.0%
Law Society of Saskatchewan	21 686	13.7%
Supreme Court of Canada Reports	14 952	9.5%
Canadian Federal Courts Reports	8 100	5.1%
Other Canadian Sources	92 496	58.7%
Total	157 697	100

Together with an experienced legal archivist, we set out to identify the decisions on CanLII [3] that featured keywords, often from Law Reports that had licensed its content to CanLII. We also contacted organizations that offered such information to members as an added value to see if they would be interested in collaborating. In the end, we gathered data from several Canadian sources (see Table 2).

3.2. Preprocessing

In order to pre-train our language model, we used the raw text (Table 3) extracted from the HTML of the 3.1M decisions, 100k commentaries, and 85k statutes and regulations in French and English languages available on CanLII. We also gathered a large collection of English language decisions with appropriate keywords, but we could not get a suitable amount of French language decisions with keywords in time for the first release. The text of every decision with keywords that we did not already have was also added to the pre-training dataset.

The extraction process separates the keywords from the rest of the text, removes summaries and cleans up the resulting text by normalizing separators and whitespaces, keeping the document structure intact.

For the pre-training denoising task, the documents were further split into individual training samples in chunks of roughly 1024 tokens. These chunks were made by cutting the text along sentence boundaries using NLTK [29]. For the fine-tuning task (keyword generation), the preprocessed text is left in a single chunk. In this step, we also normalize the keywords that were ei-

[2]Number of words per decision on the basis of NLTK.

Table 3
Pre-training Input. The </n> token is a linebreak marker leftover from whitespace normalization.

[1] </n>On May 15, 2017 the Respondent, the Vancouver</n>Park Board, passed a bylaw amen dment applicable to parks within its</n>jurisdiction, prohibiting the movement of whales to parks, the keeping of</n>whales at parks (excluding whales which were already in a park on May 15, 2017)</n>and the production or presentation in a park, of a show, performance or other</n>form of entertainment involving whales. The only park within the jurisdiction</n>of the Vancouver Park Board where whales are kept is Stanley Park, [...]

Table 4
Number of Documents in Pre-training Dataset

	Avg. Length[2]	Chunks	Count	%
Train	6529.3	4.48M	2.92M	90%
Valid	5634.8	249K	162K	5%
Test	5655.4	249K	162K	5%

ther removed from the text or supplied by an external partner to use as output targets.

The documents are then randomly shuffled and split into training (90%), validation (5%) and test (5%) sets. We also created a separate fine-tuning dataset containing the subset of decisions that had keywords. It is also split into training (90%), validation (5%) and test (5%) sets.

When pre-training multilingual models based on MBART checkpoints, we used decisions in both languages, whereas pre-training and fine-tuning for non-MBART-based models are done only on English documents (we did not have enough French documents with keywords at that time). The test sets are English-only for all models.

To avoid any information leakage, we use a bucket hashing strategy on an immutable unique index to ensure documents always end up in the same set across dataset versions. We first calculate the md5 hash of an id that is shared by every part or version of a document then convert it to an integer. We then put this id in the correct bucket by taking the modulo of the number of buckets desired (20 in our case).

This ensures that every part of a multipart document and any translation will always end up in the training set for example, or that a decision with keywords in the fine-tuning validation set will likewise be in the validation set of the pre-training dataset even if we regenerate several iterations of the datasets with new material or preprocessing.

Table 5
Documents in Fine-tuning Dataset

	Avg. Length[2]	Count	%
Train	4174.1	135K	90%
Valid	4232.9	7K	5%
Test	4172.6	7K	5%

3.3. Normalization

The keywords we had available for our training data came from very different sources that rely on very different formats. To align them more closely to the format we wanted to display to our users, we performed extensive normalization.

An issue that was immediately apparent is that some sources of keywords tend to be very terse, only two or three words, while others like the Supreme Court of Canada tend to be very long and mostly composed of descriptive sentences (Table 1). To be able to control the keyword format generated by our model and avoid having it copy whichever source format the document resembles, we introduced a keyword normalization step to our preprocessing. The complete keyword sequences were separated into a list of short keyword groups, descriptive sentences, and discussed jurisprudence and legislation. The sequence is split along the dashes into individual parts. We then use a regular expression to extract the discussed jurisprudence and legislation. Next, starting from the beginning of the sequence, we select parts of 4 words or less that do not contain a helping verb or pronoun as keywords. The rest is marked as a descriptive sentence.

One major problem that was identified when trying to generate descriptive sentences and that led us to abandon that idea was that the models were prone to hallucinating facts or subtly inverting logical propositions found in the text, thus creating believable-sounding keywords that misrepresented the decision. Removing these descriptive sentences did have some negative impact as they allow more nuanced keywords that can refer to specific facts or arguments. It also meant that we had to remove some decisions with a set of keywords that had only a single short keyword (e.g. a broad domain) followed by descriptive sentences. Once normalized, these keywords did not fit our preferred format.

In the format we settled on, the descriptive sentences in the keywords are discarded and the short keywords are limited to 6 per keyword set (a document can have several groups of keywords if it discusses various issues). The short keywords are then consolidated using a handmade mapping file of equivalent subjects to avoid having different naming conventions.

After extracting the keywords, further preprocessing is done to ensure good-quality training samples. We re-

move headers and get rid of very short decisions since those are generally assigned the same keywords (e.g. the same related lower instance decision). The various extracted keywords are also categorized into short, medium and long formats should we later decide that we do not want to use the medium-length keywords (the format chosen for the LexKey project) everywhere.

3.4. Truecasing

One major issue we faced while normalizing the keywords to create the fine-tuning dataset was that the capitalization was inconsistent: all caps for some words, title case for every word in some examples, and finally sentence case. We decided to settle on the Supreme Court of Canada's (SCC) convention of sentence case, both because it was preferred by our legal experts and because the SCC is a large, consistent and well-curated source of examples.

After trying to use standard tools such as NLTK's Part of Speech tagger which yielded poor results, we decided to fine-tune a separate version of our pre-trained lexBART model to output the proper casing on the keywords from the Supreme Court of Canada.

To avoid potential miscopying issues, we trained the model to output a token representing the proper case (lowercase, capitalized or all caps) for each word in our training and validation data. This approach yielded very good casing in the appropriate format in most cases.

Initially, we only applied truecasing to keywords from collections that were known to be badly formatted, but this preprocessing step was eventually applied to every keyword as we kept tracking down casing issues to oddities in other collections that should have been properly cased.

3.5. Hand-Curated Test Set

To validate that the models generate keywords of good quality on data from out-of-sample sources, we also had editors create a hand-labelled set of 500 documents distinct from the fine-tuning test set. They are sampled from courts and tribunals not found in any of our keyword sources. As such, none of these decisions were included in the fine-tuning dataset of the keyword generation task and we ensured that they covered topics not common in legal reports.

Editors selected them in proportions reflecting the true distribution of our complete corpus (see Table 6). To keep low-level tribunal decisions from dominating this test set, we deliberately sampled 60% of the documents from higher-level courts. The keywords assigned to these documents were also normalized, following the same steps shown earlier.

Table 6
Sources of Decisions in the Hand-Curated Test Set

Source	Proportion
Federal	12%
Alberta	12%
British Columbia	12%
Manitoba	7%
New Brunswick	7%
New Foundland	4%
Nova Scotia	8%
Northwest Territories	3%
Nunavut	2%
Ontario	18%
Prince-Edward Island	3%
Québec	0%
Saskatchewan	9%
Yukon	3%

As this hand-curating process was quite time-consuming (around 15 minutes per decision) and as the keyword-gathering process was still underway, no documents in French were included in this dataset. This meant excluding all Québec decisions for the time being.

4. Models

Given that our proof of concept used BigBirdPegasus (BBP), our first idea was to try to modify this model into a multilingual variant. BBP is warm-started from English-only RoBERTa parameters [30], then pre-trained on an unsupervised Gap Sentence Generation task [19]. Therefore, replacing the initial parameters and the tokenizer with one of the many multilingual RoBERTa-based models [31] looked feasible at first. However, after some discussion with the authors on training cost estimates, it became apparent that it was not feasible on our hardware (a small server with two A6000 GPUs) and would require renting TPUs during most of the development process.

To stay within our hardware capability, we decided to start with a pre-trained multilingual encoder-decoder model MBART50 [10, 11], further pre-train it on our data (denoising task) and then modify the encoder attention to handle longer input sequences. To speed up the development process, we did most of the experiments on a smaller English-only BART-base model first (referred to as lexBART later). This allowed us to quickly identify the effect of various preprocessing steps and find suitable hyperparameters that could then be reused on the larger MBART50 model (lexMBART).

[3]Using gradient accumulation

Table 7
Generated Keywords from Various Models on Hawes v. Redmond, 2013 NSSM 57 (CanLII)

Model	Keyword
Gold Standard	Small claims court action on unjust enrichment over dog ownership
TF-IDF	dog — gift — dogs — ownership — vet
BBP	Family Law — Child Support — Unjust Enrichment [...] Family Law — Child Support — Spousal Support — Non-compensatory Family Law — Child Support — Retroactive Support — Spousal Support Guidelines
lexMBART LSG-4k	Family law — Common-law couple — Dogs — ownership — Costs

Table 8
Hyperparameters

Hyperparameter	Value
Pre-training	
Learning Rate	1e-4
Learning Rate Schedule	Linear Decay
Optimizer	AdamW
Attention Dropout	0
Dropout	0.1
Float Type	FP16
Backend	Cuda Amp
Batch Size	64^3
Epochs	1
Steps	93 293
Eval Loss	0.4325
Fine-tuning	
Learning Rate	2e-5
Learning Rate Schedule	Linear Decay
Optimizer	AdamW
Attention Dropout	0
Dropout	0.1
LSG pool with global	True
LSG sparsity type	norm
LSG sparse block size	128
LSG sparsity factor	2
Float Type	FP16
Backend	Cuda Amp
Batch Size	16
Epochs	10
Steps	298 290
Eval Loss	0.0367
Generation	
Beams	10
Temperature	1.0
Max Length	1024

4.1. Unsupervised Pre-training

Starting from a pre-trained model checkpoint, we further pre-trained it using a denoising objective [12] on our dataset of around 3.2M legal documents (Table 4). This objective consisted of mask filling and fixing sentence permutation noise on chunks of the input documents. 15% of tokens in each sample were masked, the sentences were randomly shuffled, and the model was tasked with correctly generating the initial sample.

This pre-training was done using the standard cross-entropy loss [32] between the generated reconstituted text and the actual text before noise was applied. After more than 93k steps, the loss on the evaluation set decreases from 1.02 to 0.43. In downstream tasks, the pre-training step turned out to yield a marginal gain of up to 0.8 points for ROUGE scores, which is consistent with [33, 8]'s findings that domain adaptation is beneficial.

4.2. Handling Long Documents

To enable this model to handle long inputs, we converted its full attention layers to LSG attention [20]. The conversion from full to sparse attention makes the model less memory-greedy. LSG uses, as the name suggests, a mix of local, sparse and global attention similar to Longformer [25] along with a pooled representation of the rest of the input sequence. This allows the memory usage of attention computations to scale linearly with the input sequence length, and not quadratically like traditional transformer models.

Our pre-training objective requires that the output length is at least as long as the input length, so it is ill-suited to LSG-based models with different input and output lengths. Since the denoising task can be done with shorter document chunks and since the size mismatch does not cause problems during the fine-tuning as the keywords are always much shorter than the input text, we only modified the attention after the pre-training was done.

4.3. Supervised Fine-tuning for Keyword Generation

After pre-training, the model architecture is converted to LSG attention to allow a larger sequence input length of 4096 or 8192 tokens. It is then fine-tuned to generate the

Table 9
ROUGE Score of Various Models on Test Dataset. The top scores are in bold font and the second best are underlined.

Model	Languages	Input length	ROUGE1	ROUGE2	ROUGEL	Params	Time[4]
TF-IDF Baseline	-	-	16.0	3.6	13.5	-	-
BBP Big Patent[5]	en	4096	45.5	28.2	40.5	577M	48h
lexBART[6]	en	1024	55.7	43.6	54.4	139M	12h
lexMBART	en fr	1024	57.6	45.1	55.9	610M	55h
lexMBART LSG-4k	en fr	4096	<u>59.1</u>	<u>46.7</u>	<u>57.4</u>	618M	105h
lexMBART LSG-8k	en fr	8192	**60.1**	**47.5**	**58.2**	626M	193h

correct keywords on our labelled dataset using a cross-entropy loss. Documents exceeding the maximum input length are simply truncated by the tokenizer.

While the LSG attention can be modified to accept input lengths longer than 8k tokens, doing so proved to stretch fine-tuning time to an impractical degree. On our in-house hardware, fine-tuning an MBART50-large model extended to 4k input tokens took roughly 6 days and was proportionally longer with longer inputs (taking 12 days for 8k). In addition, 20% of our dataset is longer than 4k, and only 6% is longer than 8k.

After some experimentation, we settled on beam search as the generation strategy. We found that using 10 beams gives the best balance between generation speed and quality. We also added a rule-based post-generation processing step to remove any leftover repetitions, as beam search is prone to this issue.

5. Results

5.1. Quantitative Analysis

The order within keyword sequences matters, as terms range from the most general to the most specific legal principles. Therefore, we settled on using ROUGE [34] for assessing models' performance. As it can be seen from Table 9, all the models trained on the fine-tuning dataset outperformed both our initial BigBirdPegasus prototype and the legacy TF-IDF system on ROUGE scores by large margins of at least 10 points. The models with larger input lengths have slightly better scores with the largest model, lexMBART LSG-8k, performing best.

Table 10
ROUGE Score of Various Models on Hand-Curated Test Set. The top scores are in bold font and the second best are underlined.

Model	ROUGE1	ROUGE2	ROUGEL
TF-IDF Baseline	11.4	2.0	9.7
lexBART[6]	**32.3**	**15.3**	**30.4**
lexMBART	31.3	13.8	29.0
lexMBART LSG-4k	<u>31.8</u>	<u>14.8</u>	<u>29.8</u>
lexMBART LSG-8k	31.3	13.9	29.4

These preliminary results are however contradicted by the Hand-Curated Test Set (Table 10) where the much smaller lexBART model produced the best scores, followed by the lexMBART LSG-4k model. This suggests that models' ability to generalize to documents from unseen courts is not guaranteed to improve as the sequence input length increases. It is also possible that longer input may dilute the relevant information in cases where the model is unsure.

Table 11
ROUGE Score on Test Set by Document Length. The top scores for each length subset are in bold font.

Length	ROUGE1	ROUGE2	ROUGEL
lexBART			
< 4k tokens	56.4	44.2	55.0
> 4k tokens	52.9	41.3	50.9
> 8k tokens	49.1	36.7	46.7
lexMBART			
< 4k tokens	58.1	45.2	56.4
> 4k tokens	56.3	44.3	54.1
> 8k tokens	54.4	41.9	51.8
lexMBART LSG-4k			
< 4k tokens	59.9	47.2	58.2
> 4k tokens	57.9	**45.9**	**55.6**
> 8k tokens	55.8	43.4	53.3
lexMBART LSG-8k			
< 4k tokens	**60.4**	**47.6**	**58.6**
> 4k tokens	**58.0**	**45.9**	55.5
> 8k tokens	**57.0**	**44.6**	**54.4**

Table 11 decomposes the performance of best models depending on document lengths for the Test Set. For instance, for documents with more than 4096 and 8192 tokens (according to the MBART tokenizer), models with larger input lengths lead to better scores. The model with 8192 tokens input performed best by a small margin on documents of less than 4096 tokens and scored roughly

[4]Time taken by the fine-tuning step.
[5]Prototype trained on a single A6000 GPU. Would likely take around half the training time on 2 units.
[6]For monolingual models, French documents are removed from training and validation sets. Test set is English-only for all models.

Table 12
Impact of Preprocessing Steps on lexBART. The top scores are in bold font and the second best are underlined.

Steps	ROUGE1	ROUGE2	ROUGEL
Baseline[7]	49.0	30.7	41.2
+Pre-training	49.8	30.9	41.3
-Summaries	49.4	34.5	46.9
+Normalisation	50.2	35.6	47.8
+Truecasing	<u>50.5</u>	<u>35.7</u>	<u>48.1</u>
+Additional Sources	**55.7**	**43.6**	**54.4**

the same on those of more than 4096 but less than 8192. For longer decisions, it performed significantly better (by around 1.2 ROUGE) than the 4k model.

From Table 12, we can see the effect of each change to the model or dataset on the lexBART model that scored the best on the Hand-Curated Test Set, starting from the BART-base model. This is the same model that was used to figure out hyperparameters for the bilingual MBART models. Every step of the normalization process helped the model improve. Even removing the decision summaries from the input text only hurt the ROUGE1 score slightly. We were surprised by this low impact since we can expect summaries to contain the important information that would be found in keywords. However, all the steps related to data quality improved the ROUGE score far less than simply adding more sources of annotated decisions.

While those steps did not individually result in markedly improved metrics, their combination had a noticeable impact on the measured quality of the generation. In particular, while pre-training only improved ROUGE by 0.1 to 0.8 points, after this step, the model-generated keywords looked much more on-topic for decisions discussing subjects not found within the fine-tuning dataset (see Table 13 for an example).

5.2. Manual Qualitative Analysis

While ROUGE scores are useful when comparing models with each other, they cannot determine whether the model's outputs meet our users' quality standards. To do so, we had the final model generate keywords for the 500 documents hand-curated test set and had experts compare them to the manually generated keywords. To help them, we automatically verified if the discussed legislation could be found in the text input.

We must first emphasize that the model performed poorly on some subsets of the decisions. For example, they generate mostly random keywords when the decision is very short, but other recurring issues have been identified (see examples in Table 14). We have fixed these issues on a case-by-case basis by either excluding problematic decisions (and using the legacy TF-IDF-based keywords instead) or fixing the recurring mistakes in post-processing. The keywords displayed by the CanLII search engine can also be manually edited by an editor if required.

Table 14 shows examples of generated keywords, mostly from the hand-curated test set. In the table, Gold Standard is the keyword assigned by an expert. It is omitted when the decision is not included in any of our annotated datasets. These examples mostly stem from quality analysis done in a test environment just prior to going live. Among the other rows, TF-IDF is our legacy model, MBART50-LSG-4k is the in-production model and the Evaluator Comments are annotations added to the model output during qualitative evaluation. Some may have been translated from French.

The first example shows a marked improvement over the legacy model. The new keywords are on-topic, and cover the same subjects as the gold standard without missing important aspects. Meanwhile, the TF-IDF picks words like "recommended", "death" and "friend" that have no bearing on the case. Likewise, the second example shows another criminal case from a provincial court where the model's keywords are far better than the legacy one.

In the third, we can see one of the problematic cases where the model over-generalizes from its training data. Some tribunals that are only found in our data when their decision is appealed all have the "Judiciary review" keywords even if the decision is not being reviewed. In this case, while the model correctly identifies the topic, it incorrectly generates the "Judicial review" keywords.

The fourth, a case about compensation for a work injury had the model pick up on the description of the accident and identify it as a murder case. The model hallucinated murders in this fashion often enough that we had to keep the legacy keywords on these tribunals.

The next two are examples of very short decisions. We found the model prone to invent nonsense when processing decisions with limited context, hence our decision to not generate new keywords on decisions with less than a few paragraphs. The first of the two is only four short paragraphs, but the model does a good job of identifying the important idea. The second is a single sentence and causes the model to generate a keyword that is nearly as long and unhelpful.

Two other fairly common problem cases found during the qualitative analysis were missing topics and erroneous facts. In the missing topic example, we can see that the keywords are on-topic but that an important notion, "Evidence" is not mentioned. This was considered acceptable by reviewers. On the other hand, sometimes the model would create keywords that misrepresented the decision (in the last case, lexMBART wrongfully refers

[7]Only basic preprocessing and starting from BART-base

Table 13
Generated Keywords from Various Models on R v Woods, 2019 NWTSC 16

Model	Keywords
Gold Standard	Criminal law — Sentencing by offence — Illegal drugs — Possession for purposes of trafficking
TF-IDF	presentence report — dial-a-dope — cocaine — suspended sentence — tab
lexBART	Criminal law — Sentencing — Trafficking in drugs
lexMBART	Criminal law — Trafficking in narcotics — Statutory definition of "trafficking" — Controlled Drugs and Substances Act, s. 5(1)
lexMBART LSG-4k	Criminal law — Drug offences — Possession for purposes of trafficking
lexMBART LSG-8k	Criminal law — Sentencing — Considerations — Drug offences — Possession for purposes of trafficking — Sentencing

to "Aboriginal persons"). This kind of error can be hard to find without carefully reading the whole decision and is quite problematic to users. Thankfully, it appears to be rare enough to be acceptable in keywords labelled as automatically generated.

6. Discussion

Overall in these experiments, lexBART scored well despite its much lower parameter count. If bilingual French and English support was not a requirement down the line, its good performance would be a strong argument for picking the smaller model.

Both LSG models outperformed lexMBART but, all in all, the qualitative analysis showed only a limited difference in output quality between the lexMBART-LSG-4k and 8k models. Since the latter takes twice as long to run in both training and inference (and thus costs twice as much), we eventually decided to deploy the lexMBART-LSG-4k model to production. This necessitated adding an editorial override and keyword blacklisting feature to the publication pipeline and deploying the model using torch-serve to our AWS cloud environment, where it replaced the legacy TF-IDF system on 1.3M English language decisions from the selected courts and tribunals.

Processing those decisions took 3 days on a G5.12xlarge machine from AWS (using 4 workers, 16 threads and a batch size of 2 per GPU). The current day-to-day intake of around 600 decisions per day is handled by a G4dn.xlarge running a single worker with a batch size of one.

7. Conclusion

In this paper, we present the work done to leverage the recent advances in language modelling and generation to produce useful keywords to augment search results on the CanLII website. These efforts yielded an encoder-decoder language model named lexMBART LSG-4k (nicknamed LexKey for the sake of simplicity) that was warm-started from MBART checkpoint, further pre-trained on a large corpus of legal documents, and fine-tuned to produce structured keywords similar to those produced by legal publishers. We believe that both this model and, especially, this large multi-source corpus will allow us to continue to leverage the current NLP advances into more useful automation that previously had to be performed by hand by experts.

Despite the limitations we outlined, our custom-made keyword generator was found to perform well, generating useful keywords for a large subset of our documents. Most of the undesirable behaviours could be curbed through some post-processing and a carefully chosen keyword format.

The LexKey project started in May 2021 and in February 2023, the fine-tuned model was deployed live on CanLII on a large part of the English corpus. Both the language model and the dataset will also likely be reused in other upcoming projects like document classification. Although we cannot release the dataset because of editors' policy, we will make our pre-training and fine-tuning scripts available[8] along with the pre-trained model itself. By doing so, we also intend to showcase what is technically feasible for a small legal tech company of our size when it comes to keyword generation.

In the next development phase, we plan to source more French language decisions with keywords to provide the same feature on CanLII for both official languages (French documents were too scarce to be included in the fine-tuning dataset). We have also acquired 144K decisions from the Harvard Caselaw Access Project and 64K decisions from the Australian Federal Courts and will use them to validate whether adding data from other common law countries can help improve our model. We will also be experimenting with other legal document types like briefs or doctrines to see if our model can provide useful keywords. Finally, we intend to test the most recent large language models such as GPT-4 [23] for keyword generation.

[8] https://github.com/Lexum/lexkey-public

Table 14
Examples of Outputs From Our Models. A gold standard can cumulate two sequences of keywords. Output from TF-IDF legacy system is always lowercased. Some evaluator comments are translated from French.

Model	Keyword
Murder Case from Superior Court	
Gold Standard	Criminal law — Offences — Murder — Second degree murder — Sentencing Criminal law — Prisons — Prisoners — Releases — Parole
TF-IDF	period of parole ineligibility — offender — recommended — friends — death
lexMBART LSG-4k	Criminal law — Murder — Second degree murder — Sentencing — Parole ineligibility
Evaluator Comments	Good
Criminal Case from Provincial Court	
Gold Standard	Criminal law — Sexual offences, public morals and disorderly conduct — Sexual exploitation — Evidence Criminal law — Offences against person and reputation — Sexual assault — General offence — Evidence
TF-IDF	identification — omitted for publication — witness — sexually assaulted — don t know
lexMBART LSG-4k	Criminal law — Sexual offences — Sexual assault — Sexual touching — Evidence
Evaluator Comments	Good
Labour Relation Board	
Gold Standard	Labour and employment law — Labour law — Collective agreement — Management rights — Surveillance of employees Privacy and freedom of information — Provincial privacy legislation — Collection of personal information — Purpose and use
TF-IDF	screen captures — recording of incoming calls — requirement to record time codes — reasonableness — analysis
lexMBART LSG-4k	Labour law — Arbitration — Judicial review
Evaluator Comments	Wrong - Labour law — Arbitration is good but Judicial review is wrong.
Worker Compensation Board	
TF-IDF	knee — worker — pre-existing degenerative changes — pre-existing condition — aggravation
lexMBART LSG-4k	Criminal law — Murder — Second degree murder — Evidence — Identification
Evaluator Comments	Wrong - about workplace injury, not murder
Short Decision About a Fine	
TF-IDF	complied — fine — imprisoned — merits — varying
lexMBART LSG-4k	Practice and procedure — Fine — Compliance with judgment
Evaluator Comments	Good
Very Short Decision	
Whole Text	The Reasons for Judgment rendered in file T-1291-97 apply to the appellant in this file.
TF-IDF	file — rendered — apply
lexMBART LSG-4k	Practice — Judgments and orders — Reasons for judgment — Application to vary
Wrong Output	
Gold Standard	Labour and employment law — Labour law — Unfair labour practices — Employer practices — Interference with union activities Labour and employment law — Labour law — Unfair labour practices — Remedies — Miscellaneous
TF-IDF	vote — scheduled — unfair labour practice — shift — hours
lexMBART LSG-4k	Labour relations — Certification — Wishes and preferences — Employee vote
Evaluator Comments	Wrong - ”Wishes and preferences” is wrong. Also incomplete.
Missing Topic	
Gold Standard	Criminal law — Offences against person and reputation — Sexual assault — General offence — Evidence Evidence — Hearsay — Traditional exceptions to rule against admission — Spontaneous statements
TF-IDF	audio recordings — testimony — evidence — sexual assault — witness
lexMBART LSG-4k	Criminal law — Sexual offences — Sexual assault — Offences against persons — Unlawful confinement
Evaluator Comments	Not Bad - No mention of Evidence
Factually Wrong	
Gold Standard	Criminal law — Offences — Robbery — Sentencing — Adult offenders
TF-IDF	sentence — pre-sentence report — robberies — community — offences
lexMBART LSG-4k	Criminal law — Property offences — Robbery — Sentencing — Aboriginal persons
Evaluator Comments	Wrong - par. 40 “This offender is not aboriginal.”

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[2] J. Devlin, M.-W. Chang, K. Lee, 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), 2019, pp. 4171–4186.

[3] Canadian Legal Information Institute, https://www.canlii.org/en/, 2001. Accessed: 2023-02-05.

[4] S. C. of Canada | Cour suprême du Canada, Style Manual | Guide de rédaction, Supreme Court of Canada | Cour suprême du Canada, 1987.

[5] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://aclanthology.org/2020.emnlp-demos.6. doi:10.18653/v1/2020.emnlp-demos.6.

[6] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, et al., Big bird: Transformers for longer sequences., in: NeurIPS, 2020.

[7] E. Sharma, C. Li, L. Wang, BIGPATENT: A large-scale dataset for abstractive and coherent summarization, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 2204–2213. URL: https://aclanthology.org/P19-1212. doi:10.18653/v1/P19-1212.

[8] N. Garneau, E. Gaumond, L. Lamontagne, P.-L. Déziel, Criminelbart: A french canadian legal language model specialized in criminal law, in: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, 2021, pp. 256–257.

[9] S. Douka, H. Abdine, M. Vazirgiannis, R. El Hamdani, D. R. Amariles, Juribert: A masked-language model adaptation for french legal text, in: Proceedings of the Natural Legal Language Processing Workshop 2021, 2021, pp. 95–101.

[10] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, L. Zettlemoyer, Multilingual denoising pre-training for neural machine translation, Transactions of the Association for Computational Linguistics 8 (2020) 726–742.

[11] Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu, A. Fan, Multilingual translation with extensible multilingual pretraining and finetuning (2020). arXiv:2008.00401.

[12] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. URL: https://aclanthology.org/2020.acl-main.703. doi:10.18653/v1/2020.acl-main.703.

[13] J. Ramos, Using tf-idf to determine word relevance in document queries, 1999.

[14] A. Farzindar, G. Lapalme, Letsum, an automatic legal text summarizing, in: Legal knowledge and information systems: JURIX 2004, the seventeenth annual conference, volume 120, IOS Press, 2004, p. 11.

[15] B. Hachey, C. Grover, Extractive summarisation of legal texts, Artificial Intelligence and Law 14 (2006) 305–345.

[16] S. Polsley, P. Jhunjhunwala, R. Huang, CaseSummarizer: A system for automated summarization of legal texts, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, The COLING 2016 Organizing Committee, Osaka, Japan, 2016, pp. 258–262. URL: https://aclanthology.org/C16-2054.

[17] D. Miller, Leveraging bert for extractive text summarization on lectures, 2019. URL: https://arxiv.org/abs/1906.04165. doi:10.48550/ARXIV.1906.04165.

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, The Journal of Machine Learning Research 21 (2020) 5485–5551.

[19] J. Zhang, Y. Zhao, M. Saleh, P. Liu, Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, in: International Conference on Machine Learning, PMLR, 2020, pp. 11328–11339.

[20] C. Condevaux, S. Harispe, Lsg attention: Extrapolation of pretrained transformers to long sequences, in: Advances in Knowledge Discovery and Data Mining: 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2023, Osaka, Japan, May 25–28, 2023, Proceedings, Part I, Springer, 2023, pp. 443–454.

[21] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Ka-

plan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[22] OpenAI, Introducing ChatGPT, https://openai.com/blog/chatgpt, 2022. Accessed: 2023-02-05.

[23] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.

[24] T. Dao, D. Fu, S. Ermon, A. Rudra, C. Ré, Flashattention: Fast and memory-efficient exact attention with io-awareness, Advances in Neural Information Processing Systems 35 (2022) 16344–16359.

[25] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, 2020. URL: https://arxiv.org/abs/2004.05150. doi:10.48550/ARXIV.2004.05150.

[26] M. Guo, J. Ainslie, D. Uthus, S. Ontanon, J. Ni, Y.-H. Sung, Y. Yang, LongT5: Efficient text-to-text transformer for long sequences, in: Findings of the Association for Computational Linguistics: NAACL 2022, Association for Computational Linguistics, Seattle, United States, 2022, pp. 724–736. URL: https://aclanthology.org/2022.findings-naacl.55. doi:10.18653/v1/2022.findings-naacl.55.

[27] I. Chalkidis, X. Dai, M. Fergadiotis, P. Malakasiotis, D. Elliott, An exploration of hierarchical attention transformers for efficient long document classification, 2022. URL: https://arxiv.org/abs/2210.05529. doi:10.48550/ARXIV.2210.05529.

[28] Z. Shen, K. Lo, L. Yu, N. Dahlberg, M. Schlanger, D. Downey, Multi-lexsum: Real-world summaries of civil rights lawsuits at multiple granularities, in: Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2022. URL: https://openreview.net/forum?id=z1d8fUiS8Cr.

[29] S. Bird, E. Loper, NLTK: The natural language toolkit, in: Proceedings of the ACL Interactive Poster and Demonstration Sessions, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 214–217. URL: https://aclanthology.org/P04-3031.

[30] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: https://arxiv.org/abs/1907.11692. doi:10.48550/ARXIV.1907.11692.

[31] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8440–8451.

[32] I. J. Good, Rational decisions, Journal of the Royal Statistical Society. Series B (Methodological) 14 (1952) 107–114. URL: http://www.jstor.org/stable/2984087.

[33] L. Zheng, N. Guha, B. R. Anderson, P. Henderson, D. E. Ho, When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings, in: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, 2021, pp. 159–168.

[34] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.