

# Automated Refugee Case Analysis: An NLP Pipeline for Supporting Legal Practitioners

Claire Barale and Michael Rovatsos

School of Informatics

The University of Edinburgh

{claire.barale,michael.rovatsos}@ed.ac.uk

Nehal Bhuta

School of Law

The University of Edinburgh

nehal.bhuta@ed.ac.uk

## Abstract

In this paper, we introduce an end-to-end pipeline for retrieving, processing, and extracting targeted information from legal cases. We investigate an under-studied legal domain with a case study on refugee law in Canada.

Searching case law for past similar cases is a key part of legal work for both lawyers and judges, the potential end-users of our prototype. While traditional named-entity recognition labels such as dates provide meaningful information in legal work, we propose to extend existing models and retrieve a total of 19 useful categories of items from refugee cases.

After creating a novel data set of cases, we perform information extraction based on state-of-the-art neural named-entity recognition (NER). We test different architectures including two transformer models, using contextual and non-contextual embeddings, and compare general purpose versus domain-specific pre-training.

The results demonstrate that models pre-trained on legal data perform best despite their smaller size, suggesting that domain matching had a larger effect than network architecture. We achieve a F1 score above 90% on five of the targeted categories and over 80% on four further categories.

## 1 Introduction

The retrieval of similar cases and their analysis is a task at the core of legal work. Legal search tools are widely used by lawyers and counsels to write applications and by judges to inform their decision-making process. However, this task poses a series of challenges to legal professionals: (i) it is an expensive and time-consuming task that accounts for 30% of the legal work on average (Poje, 2014), (ii) databases can be very large, with legal search tools gathering billions of documents, and (iii) selection of cases can be imprecise and may return many irrelevant cases, which creates the need to read more cases than necessary.

In Canada, from the date of the first claim to the final decision outcome, a claimant can expect to wait 24 months for refugee claims and 12 months for refugee appeals<sup>1</sup>. Long processing times are due to a significant backlog and to the amount of work required from counsels that help claimants file their claims, and who are frequently legal aid or NGO employees.

We find that these challenges are well-suited for NLP-based solutions and investigate the feasibility of automating all steps of the legal search for past similar cases. We construct an end-to-end pipeline that aims at facilitating this multi-step process, thereby supporting and speeding up the work of both lawyers and judges in *Refugee Status Determination (RSD)*. We provide a level of granularity and precision that goes beyond that of existing legal search tools such as *Westlaw*, *LexisNexis*, or *Refworld*<sup>2</sup> (Custis et al., 2019), which operate at the document level. *Refworld* is an online database maintained by the United Nations which helps retrieve relevant precedent cases and legislation. However, the level of precision with which one can search for cases is limited. Moreover, our pipeline guarantees increased transparency, enabling end users to choose the criteria of legal search they find most relevant to their task among the proposed categories that act as filters for a search.

Specific literature studying refugee law and AI is sparse. Attention has been given to the classification and prediction of asylum cases in the United States (Chen and Eigel, 2017; Dunn et al., 2017). On Canadian data, research has been conducted to analyze the disparities in refugee decisions using statistical analysis (Rehaag, 2007, 2019; Cameron et al., 2021). However, those studies rely mostly on tabular data. We propose to work directly on

<sup>1</sup>Wait times for refugee claims in Canada: <https://irb.gc.ca/en/transparency/pac-binder-nov-2020/Pages/pac8a.aspx?undefined&wbdisable=true>

<sup>2</sup>Refworld: <https://www.refworld.org/>

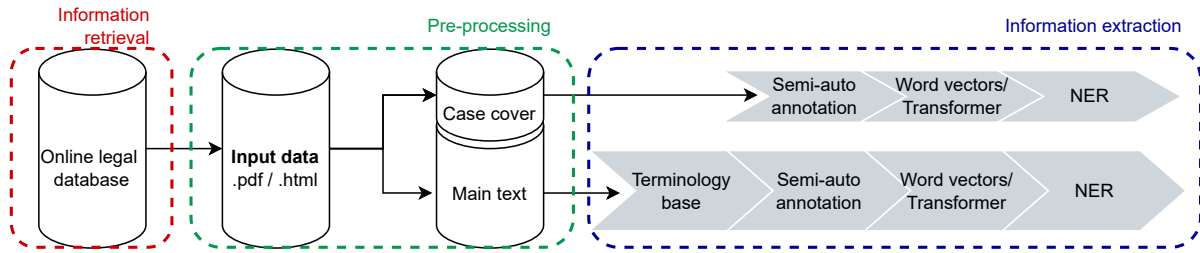


Figure 1: End-to-end automated pipeline

the text of refugee cases. To the best of our knowledge, no previous work implements an end-to-end pipeline and state-of-the-art NLP methods in the field of refugee law.

We provide an NLP-based end-to-end prototype for automating refugee case analysis built on historical (already decided) cases, which are currently available only in unstructured or semi-structured formats, and which represent the input data to our pipeline. The end goal of our approach is to add structure to the database of cases by extracting targeted information described in table 1 from the case documents, and providing the results in a structured format to significantly enrich the search options for cases. Thereby, the input data set of cases is described in a structured manner based on our extracted categories of items, adding extensive capabilities for legal search.

The pipeline described in figure 1 begins by searching and downloading cases (information retrieval, paragraph 4.1), pre-processing them (paragraph 4.2), extracting items previously identified as relevant by legal professionals. It then outputs a structured, precise database of refugee cases (information extraction, paragraph 4.3). In the information extraction step, we test different training and pre-training architectures in order to determine the best methods to apply to the refugee case documents. We construct each step with the aim of minimizing the need for human effort in creating labeled training data, aiming to achieve the best possible accuracy on each extracted information item. We discuss technical choices and methodologies in section 5. Finally, we evaluate the information extraction step on precision, recall, and F1 score, and present detailed results in section 6.

We demonstrate that annotation can be sped up by the use of a terminology base while incorporating domain knowledge and semi-automated annotation tools. We find that domain matching is important for training to achieve the highest possible

scores. We reach satisfactory token classification results on a majority of our chosen categories. The contributions of this paper are as follows:

1. First, we retrieve 59,112 historic decision documents (dated from 1996 to 2022) from online services of the Canadian Legal Information Institute (CanLII) based on context-based indexing and metadata to curate a collection of federal *Refugee Status Determination (RSD)* cases. Our automated retrieval process is exhaustive and comprises all available cases. It is superior to human-based manual retrieval in terms of error proneness and processing time.
2. Second, we proposed an information extraction pipeline that involves pre-processing, construction of a terminology base, labeling data, and using word vectors and NER models to augment the data with structured information. We fine-tune state-of-the-art neural network models to the corpus of our retrieved cases by training on newly created gold-standard text annotations specific to our defined categories of interest.
3. Lastly, we extract the targeted category items from the retrieved cases and create a structured database from our results. We introduce structure to the world of unstructured legal *RSD* cases and thereby increase the transparency of stated legal grounds, judge reasoning, and decision outcomes across all processed cases.

## 2 Background and motivation

At the core of the ongoing refugee crisis is the legal and administrative procedure of *Refugee Status Determination (RSD)*, which can be summarized in three sub-procedures: (i) a formal claim for refugee protection by a claimant who is commonly supported by a lawyer, (ii) the decision-making

	Label	A	Description	Example
Case cover	<i>General</i>			
	DATE	1,219	absolute or relative dates or periods	date of the hearing and date of the decision
	GPE	871	cities, countries, regions	place of the hearing
	ORG	278	tribunals	"immigration appeal division", "refugee protection division"
	PERSON	119	names	name of the panel and counsels
Main text	<i>Information on claimant and allegations</i>			
	CLAIMANT_EVENT	1,575	verbs or nouns describing an event of the story of the claimant	"rape", "threat", "attacks", "fled"
	CLAIMANT_INFO	235	age, gender, citizenship, occupation	"28 year old", "citizen of Iran", "female"
	GPE	732	cities, countries, regions	countries of past residency or places of hearings: "toronto, ontario"
	NORP	129	nationalities, religious, political or ethnic groups or communities	"hutu", "nigerian", "christian"
	<i>Legal procedure</i>			
	ORG	549	tribunals, NGOs, companies	"human rights watch", "refugee protection division"
	PROCEDURE	594	steps in the claim and legal procedure events	"removal order", "sponsorship for application"
	<i>Analysis and reasons for decision outcome</i>			
	CREDIBILITY	684	mentions of credibility in the determination	"lack of evidence", "inconsistencies"
	DETERMINATION	76	outcome of the decision (accept/reject)	"appeal is dismissed", "panel determines that the claimant is not a convention refugee"
	DOC_EVIDENCE	768	pieces of evidence, proofs, supporting documents	"passport", "medical record", "marriage certificate"
	EXPLANATION	404	reasons given by the panel for the determination	"fear of persecution", "no protection by the state"
	<i>Timeline</i>			
	DATE	628	absolute or relative dates or periods	"for two ears", "june, 4th 1996"
<i>Names</i>				
PERSON	154	names	claimants' names, their family, name of judges	
<i>Citations</i>				
LAW	476	legislation and international conventions	state law and international conventions, "section 1(a) of the convention"	
LAW_CASE	109	case law and past decided cases, by the same tribunal or another	"xxx v. minister of canada, 1994"	
LAW_REPORT	18	country reports written by NGOs or the United Nations	"amnesty international, surviving death: police and military torture of women in mexico, 2016"	

Table 1: Targeted categories (**Label**) for extraction with number of annotations (**A**) per label, sorted alphabetically

process of a panel of judges and (iii) the final decision outcome with written justification for granting refugee protection or not.

Refugee protection decisions are high-stakes procedures that target 4.6 million asylum seekers worldwide as of mid-2022. In Canada alone, 48,014 new claims and 10,055 appeals were filed in 2021<sup>3</sup>. As stated in the introduction, processing times of refugee claims vary and range from a few months to several years. One of the reasons for the long processing times is the effort required for similar cases search. Case research is an essential part of the counsel's work in preparation for a new claim file. This search involves retrieving citations and references to previous, ideally successful RSD cases that exhibit similarities to the case in preparation such as the country of origin or the reason for the claim. Equally, judges rely on researching previous cases to justify their reasoning and ensure coherency across rulings.

While each case exhibits individual characteristics and details, legal practitioners typically search for similarities based on the constitution of the panel, the country of origin and the characteristics of the claimant, the year the claim was made in relation to a particular geopolitical situation, the legal procedures involved, the grounds for the decision, the legislation, as well as other cases or reports that are cited.

Our work aims to support legal practitioners, both lawyers preparing the application file and judges having to reach a decision, by automating

the time-consuming search for similar legal cases referred to here as *refugee case analysis*. As a case study, we work on first instance and appeal decisions made by the *Immigration and Refugee Board of Canada*. A common approach used by legal practitioners is to manually search and filter past RSD cases on online services such as CanLII or Refworld by elementary *document text* search, which is a keyword-based *find exact* search, or by date.

Our defined categories of interest are described in table 1. The labels have been defined and decided upon with the help of three experienced refugee lawyers. From the interviews, we curated a list of keywords, grounds, and legal elements determining a decision. Moreover, we analyzed a sample of 50 Canadian refugee cases recommended by the interviewees to be representative over years of the claim and tribunals.

We use the pre-defined labels provided by spaCy's state-of-the-art `EntityRecognizer` class including DATE, PERSON, GPE, ORG, NORP, LAW and extend this list with new additional labels that we created and trained from scratch.

Each case document comprises a *case cover* page (the first page) and the *main text* which differ in the type and format of their information content. Therefore, we chose separate labels for the *case cover*. The *case cover* contains general information about the case (cf. example in Appendix A). While the main text is presented as a full-body text, the *case cover* page consists of semi-structured information which could that could be roughly described as tabular, except it does not follow a clear

<sup>3</sup><https://irb.gc.ca/en/statistics/Pages/index.aspx>

layout. Based on the *case cover* page we aim to extract meta-information about each claim using four labels (table 1).

For the *main text*, we chose 15 labels that represent characteristics reflective of similarity among different cases. To link cases to each other and later facilitate similar case retrieval, we also extract three categories of citations i.e. LAW for legal texts, LAW\_CASES for other mentioned past cases, and LAW\_REPORT for external sources of information such as country reports. Additionally, the CREDIBILITY label retrieves mentions made of credibility concerns in the claimant’s allegations, which tends to be among the deciding factors for the success of a claim and is hence essential to understand the reasoning that led to the legal determination at hand.

A successful implementation of a system capable of extracting this information reliably would provide several benefits to legal practitioners: (i) facilitating, speeding up, and focusing legal search, (ii) reducing the time spent on a claim and on providing relevant references, potentially resulting in a file that has more chances of being accepted, and (iii) for judges, to ensure consistent outcomes across time and different jurisdictions or claimant populations.

### 3 Research approach

Our approach is guided by investigating the hypothesis that NER methods can be used to extract structured information from legal cases, i.e. we want to determine whether state-of-the-art methods can be used to improve the transparency and processing of refugee cases. Consistency of the decision-making process and thorough assessment of legal procedure steps are crucial aspects ensuring that legal decision outcomes are transparent, high-quality, and well-informed. Consequently, key research questions we need to address include:

**Training data requirements** How many labeled samples are needed? Can keyword-matching methods or terminology databases be leveraged to reduce the need for human annotation?

**Extraction** What methods are best suited to identify and extract the target information from legal cases?

**Replicability** To what extent might our methods generalize to other legal data sets (other legal fields or other jurisdictions)?

**Pre-training** How important is the pre-training

step? How important is *domain matching*: do domain-specific pre-training perform better than general-purpose embeddings, despite their smaller sizes?

**Architectures** How important is the architecture applied to the information extraction tasks, in terms of F1 score, precision, and recall?

## 4 Pipeline details and experimental setup

In this section, we detail each step of the pipeline as presented in figure 1 and how it compares to the current legal search process. Subsequently, in 5 we describe the training data creation process, and the network architectures tested. The code for our implementation and experiments can be found at [https://github.com/clairebarale/refugee\\_cases\\_ner](https://github.com/clairebarale/refugee_cases_ner).

### 4.1 Information retrieval: case search

We retrieve 59,112 cases processed by the *Immigration and Refugee Board* of Canada that range from 1996 to 2022. The case documents have been collected from CanLII in two formats, PDF and HTML. The CanLII web interface serves queries through their web API accessible at the endpoint with URL <https://www.canlii.org/en/search/ajaxSearch.do>. For meaningful queries, the web API exposes a number of HTTP-GET request parameters and corresponding values which are to be appended to the URL but preceded by a single question mark and then concatenated by a single ampersand each. For instance, in the parameter=value pairs in the following example, the keyword search exactly matches the text *REFUGEE*, and we retrieve the second page of a paginated list of decisions from March 2004 sorted by descending date, which returns a JSON object (Full query: [https://www.canlii.org/en/search/ajaxSearch.do?type=decision&ccId=cisr&text=EXACT\(REFUGEE\)&startDate=2004-03-01&endDate=2004-03-31&sort=decisionDateDesc&page=2](https://www.canlii.org/en/search/ajaxSearch.do?type=decision&ccId=cisr&text=EXACT(REFUGEE)&startDate=2004-03-01&endDate=2004-03-31&sort=decisionDateDesc&page=2)). Note that CanLII applies pagination to the search results in order to limit the size of returning objects per request.

### 4.2 Preprocessing

We obtain two sets: (1) a set of *case covers* that consists of semi-structured data and displays meta-information and (2) a set of *main text* that contains the body of each case, in full text.

Generally, the CanLII database renders the decision case documents as HTML pages for display in modern web browsers but also provides PDF files. We use PyPDF2<sup>4</sup> for parsing the contents of PDF files as text. To parse the contents of HTML files as text input to our NLP pipeline, we use the BeautifulSoup<sup>5</sup> python library.

The choice between PDF and HTML format is based on multiple reasons, as each format has its own advantages and disadvantages. First, depending on the text format PyPDF2 occasionally adds excessive white space between letters of the same word. Also, the PDF document is parsed line-by-line from left to right, top to bottom. Therefore, multi-column text is often mistakenly concatenated as a single line of text. However, the available PDF documents are separated by pages and PyPDF2 provides functionality to select individual document pages which we used to select the case cover page that provides case details for each document. HTML as markup language provides exact anchors with HTML tags, which, in most cases, are denoted by opening and closing tag parts such as <p> and </p> for enclosing a paragraph.

When processing the *main text* of each case document, we parse the HTML files using BeautifulSoup, remove the case cover to keep only the full-body text, and tokenize the text by sentence using the NLTK<sup>6</sup>. Our preference to tokenize by sentence facilitates the annotation process while keeping the majority of the context. We also experimented with splitting by paragraph which yielded relatively large chunks of text, whereas splitting by phrase did not keep enough context during the annotation process. To gather results, we create a pandas DataFrame, create a sentence per row, and save it to a CSV file.

For the *case cover*, we exploit PyPDF2's functionality to extract the text of the first page from the PDF format. In contrast to this, when using BeautifulSoup we could not rely on HTML tags (neither through generic tag selection nor by CSS identifier (ID) or CSS class), to retrieve the first page of the document robustly. After extracting this page for each case, we parse the PDF files as plain text. Combined with the metadata from the document retrieval provided by CanLII, we derive

<sup>4</sup>PyPDF2: <https://github.com/py-pdf/PyPDF2>

<sup>5</sup>BeautifulSoup: <https://beautiful-soup-4.readthedocs.io/en/latest/>

<sup>6</sup>NLTK tokenizer: <https://www.nltk.org/api/nltk.tokenize.html>

the case identifier number and assign it to the corresponding PDF file. As a next step and similar to the procedure for the main body of each document, we create a pandas DataFrame from the extracted data and save it as a CSV file with case identifier numbers and their associated case cover.

For both file formats, we perform basic text cleaning, converting letters to lowercase, and removing excessive white space and random newlines.

### 4.3 Information extraction

The goal of our pipeline is not only to retrieve the cases but to structure them with a high level of precision and specificity, and to output a tabular file where each column stores specific information of each of our target types for each case. Using such structured information, legal practitioners can find similar cases with ease instead of reading carefully through several cases before finding a few cases similar to their own cases by selecting attributes in one or several of the extracted categories.

We chose to use neural network approaches to perform the information extraction step. After some experimentation, approaches such as simple matching and regular expressions search proved too narrow and unsuitable for our data. Given the diversity of formulations and layouts, phrasing that captures context is quite important. Similarly, we discard unsupervised approaches based on the similarity of the text at the document or paragraph level because we favor transparency to the end user in order to enable leveraging legal practitioners' knowledge and expertise.

Extraction of target information can be done using sequence-labeling classification. NER methods are well-suited to the task of extracting keywords and short phrases from a text. To this end, we create a training set of annotated samples as explained in the next section 5.1. Labeled sentences are collected in jsonlines format, which we convert to the binary spaCy-required format and use as training and validation data for our NER pipeline.

## 5 Methodology

### 5.1 Training data creation

We choose to use a machine learning component for text similarity to reinforce the consistency of the annotations. In line with our previous step of pre-processing, we annotate the case cover section and the main text separately. While we decided

to annotate the whole page of the case cover because the semi-structured nature of the text makes tokenization approximate, we perform annotation of the main text as a sentence-based task, preserving some context. We use the Prodigy annotation tool<sup>7</sup>, which provides semi-automatic annotations and active learning in order to speed up and improve the manual labeling work in terms of consistency and accuracy of annotation. We convert the two pandas Dataframes containing the pre-processed text to jsonlines which is the preferred format for Prodigy. We annotate 346 case covers and 2,436 sentences for the main text, which are chosen from the corpus at random.

To collect annotated samples on traditional NER labels (DATE, ORG, GPE, PERSON, NORP, LAW), we use suggestions from general purpose pre-trained embeddings<sup>8</sup>. For the remaining labels (CLAIMANT\_INFO, CLAIMANT\_EVENT, PROCEDURE, DOC\_EVIDENCE, EXPLANATION, DETERMINATION, CREDIBILITY), and still with the aim of improving consistency of annotation, we create a terminology base (as shown on pipeline description figure 1). At annotation time, patterns are matched with shown sentences. The human annotator only corrects them, creating a gold standard set of sentences and considerably speeding up the labeling task.

To create a terminology base for each target category, we first extract keywords describing cases from CanLII metadata retrieved during the information retrieval step. To this initial list of tokens, we add a list of tokens that were manually flagged in cases by legal professionals. We delete duplicates and some irrelevant or too general words such as “claimant” or “refugee”, and manually assign the selected keywords to the appropriate label to obtain a list of tokens and short phrases per label. In order to extend our terminology base, we use the sense2vec model<sup>9</sup>(based on word2vec (Mikolov et al., 2013)) to generate similar words and phrases. We select every word that is at least 70% similar to the original keyword in terms of cosine similarity and obtain a JSON file that contains 1,001 collected patterns. This method allows us to create a larger number of labeled data compared to fully manual annotation in the same amount of time.

Table 1 describes the breakdown of labels in

<sup>7</sup>Prodigy: <https://prodi.gy/docs>

<sup>8</sup><https://spacy.io/models/en>

<sup>9</sup>Sense2vec:

<https://github.com/explosion/sense2vec>

our annotated data. There is a clear imbalance across categories of items, with some labels being infrequent (NORP, DETERMINATION, PERSON, LAW\_REPORT, LAW\_CASE). Some labels are present very few times per case: DETERMINATION occurs only once per case, PERSON does not occur frequently since most cases are anonymized.

## 5.2 Experimental conditions and architectures

**Train, dev, test split** We trained the NER models using 80% of the labeled data as our training set (276 case covers and 1,951 sentences for the main text, respectively), 10% of the labeled data as our development set (35 case covers and 244 sentences) and 10% of the labeled data as the test set for evaluation (35 case covers and 244 sentences).

### Pre-training static and contextual embeddings

As the first layer of the NER network, we add pre-trained character-level embeddings in order to isolate the effect of pre-training from the effect of the architecture and improve the F1 score on target items. We fine-tune GloVe vectors ((Pennington et al., 2014), 6B tokens, 400K vocabulary, uncased, 50 dimensions) on our data using the Mittens<sup>10</sup> python package (Dingwall and Potts, 2018) and create 970 static vectors. On top of the generated static vectors, we add dynamic contextualized vectors using pre-training embeddings based on BERT (J. Devlin and Toutanova, 2019), updating weights on our corpus of cases. Because the text of the case cover is presented in a semi-structured format, we consider that it is unnecessary to perform pre-training given the lack of context around the target items.

**Architectures** We experiment with five different architectures on the case cover and seven different architectures on the main text: five based on convolutional neural networks (CNN) using different word embeddings and two transformer architectures. We train a CNN without added vectors as a baseline. Only the transformer architectures require training on a GPU. We use the spaCy pipelines<sup>11</sup> (tokenizer, CNN and transformer) and the HuggingFace datasets<sup>12</sup>. All CNNs use an

<sup>10</sup>Mittens: <https://github.com/roamanalytics/mittens>

<sup>11</sup>SpaCy: <https://spacy.io/api/entityrecognizer>

<sup>12</sup>roBERTa: <https://huggingface.co/roberta-base>, LegalBERT: <https://huggingface.co/nlpaueb/>

for these reasons, i confirm the rpd's determination DETERMINATION, namely that  
 xxxx xxxx, xxxx xxxx, xxxx xxxx, xxxx xxxx, xxxx xxxx and xxxx xxxx xxxx are neith  
 er "convention refugees" DETERMINATION under section 96of theirpa LAW, nor "pe  
 rsons in need of protection" within DETERMINATION the meaning of section 97of the  
 irpa LAW.

chavez, rodrigo v.

m.c.i.

(iad ta3-24409), hoare, january 17, 2005 at paragraph 3

Figure 2: Example of an error in tokenization

Adam optimizer function. Since the sentence-labeling task is well-suited to the masked language modeling objective, we chose to experiment with roBERTa (Liu et al., 2019) and LegalBERT (Chalkidis et al., 2020) in order to compare performance between a general content and a legal content model.

We train separately on the case cover, the traditional NER labels (GPE, NORP, ORG, DATE, PERSON, LAW), and the labels we created from scratch since it was observed that labels trained from scratch benefit from a lower learning rate (0.0005 versus 0.001 for the traditional labels).

## 6 Results and evaluation

Our experimental results are presented in table 2 in absolute terms and relative to the baseline in figure 3 below. Our chosen baseline is a CNN with no additional vectors. We present them per label because of the disparities in the scores. The upper rows contain results on the case cover and the lower rows results on the main text. The evaluation metrics applied serve a dual purpose: for future research, achieving a high F1 score and precision-recall balance is key, while for our potential legal end users we assume that the recall measure is much more important as it measures how many of the predicted entities are correct.

For the case cover, we obtain satisfactory results on all labels with F1 scores above 90% for three of them and 84.78% for name extraction. Apart from names, CNN architectures perform better, with dates achieving the highest score with randomly initialized embeddings. We explain this with the specific layout of this page (Annex A). The only gain of using a transformer-based model is to achieve a higher recall compared to the CNN-based architectures.

For the main text, results vary across labels: we obtain a score above 80% for DATE, GPE, PERSON, ORG with the best score on roBERTa, but

legal-bert-base-uncased

scores lower than 60% on EXPLANATION, LAW, LAW\_CASE. Overall, when using transformers, we observe a better precision-recall balance.

Results on three labels DETERMINATION, LAW\_REPORT, NORP are unreliable because of the limited sample both for training and testing. DETERMINATION appears only once per case, and LAW\_REPORT appears in a few cases only. Further annotation would require selecting the paragraphs of cases where these items appear to augment the size of the sample. We leave this task to future work.

Explanations for other low scores are partly to be found in the tokenization errors reported during the human-labeling task. Figure 2 shows an example of wrong tokenization on two categories LAW and LAW\_CASE for which we believe bad tokenization is the primary explanation for low scores (similarly reported by Sanchez). In the first sentence of the figure, words are not correctly split between “under” and “section” and between the section number and “of”. On the lower part of the figure, sentence tokenization does not correctly split the case reference as it is confused by the dot present in the middle. In this example, the case name is displayed as three different sentences, making the labeling task impossible.

The most appropriate pre-training varies across labels: For categories on which CNN performs best such as CREDIBILITY, DOC\_EVIDENCE, LAW, we find that fine-tuning static vectors performs better than randomly initialized embeddings or dynamic vectors, which suggests that context was not essential when retrieving spans of text (pre-training relies on tri-grams). This could derive from the methods of annotation that were terminology-based for those labels. While the target items may contain particular vocabulary such as “personal information form” for DOC\_EVIDENCE, context is of minimal importance since those phrases would not appear in another context or under another label. On the contrary, context seems much more important for retrieving procedural steps (PROCEDURE), which is the only category where the pre-training layer with contextual embeddings significantly increases the F1 score.

In the majority of categories, we find that the content of the pre-training is important (CLAIMANT\_EVENT, CREDIBILITY, DATE, DOC\_EVIDENCE, EXPLANATION, LAW, PROCEDURE). Results show that domain-specific

Label \ Architecture	baseline			CNN+rsv			CNN+fTs			CNN+rsv+pt			CNN+fTs+pt			RoBERTa			LegalBERT		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
<b>Header and cover page</b>																					
DATE	97.46	95.04	96.23	95.90	96.69	<b>96.30</b>	-	-	-	-	-	-	-	-	-	89.92	89.17	89.54	89.08	88.33	88.70
GPE	92.96	91.67	<b>92.31</b>	90.14	88.89	89.51	-	-	-	-	-	-	-	-	-	90.54	93.06	91.78	88.00	91.67	89.80
ORG	94.74	90.00	<b>92.31</b>	94.74	90.00	<b>92.31</b>	-	-	-	-	-	-	-	-	-	79.17	95.00	86.36	86.36	95.00	90.48
PERSON	80.80	84.17	82.45	81.75	85.83	83.74	-	-	-	-	-	-	-	-	-	75.48	96.69	<b>84.78</b>	69.82	97.52	81.38
<b>Main text document body</b>																					
CLAIMANT_EVENT	60.36	44.67	51.34	57.02	46.00	50.92	57.89	36.67	44.90	55.04	47.33	50.90	63.71	52.67	57.66	64.34	61.33	62.80	65.10	64.67	<b>64.88</b>
CLAIMANT_INFO	55.00	61.11	57.89	47.83	61.11	53.66	61.11	61.11	61.11	55.56	55.56	55.56	63.16	66.67	<b>64.86</b>	63.16	66.67	<b>64.86</b>	57.89	61.11	59.46
CREDIBILITY	68.57	50.00	57.83	62.50	52.08	56.82	69.23	56.25	<b>62.07</b>	74.19	47.92	58.23	68.42	54.17	60.47	56.60	62.50	59.41	62.50	52.08	56.82
DATE	72.34	69.39	70.83	94.44	69.39	80.00	72.34	69.39	70.83	81.40	71.43	76.09	83.33	71.43	76.92	85.11	81.63	83.33	86.96	81.63	<b>84.21</b>
DETERMINATION	100.00	36.36	53.33	85.71	54.55	<b>66.67</b>	100.00	36.36	53.33	83.33	45.45	58.82	85.71	54.55	66.67	83.33	45.45	58.82	42.86	27.27	33.33
DOC_EVIDENCE	77.61	74.29	75.91	77.27	72.86	75.00	80.60	77.14	<b>78.83</b>	75.00	72.86	73.91	68.42	74.29	71.23	67.53	74.29	70.75	71.62	75.71	73.61
EXPLANATION	46.00	43.40	44.66	60.98	47.17	53.19	56.82	47.17	51.55	58.14	47.17	52.08	53.49	43.40	47.92	54.17	49.06	51.49	60.47	49.06	<b>54.17</b>
GPE	88.76	89.77	89.27	90.59	87.50	89.02	91.57	86.36	<b>88.89</b>	90.48	86.36	88.37	89.29	85.23	87.21	95.35	93.18	<b>94.25</b>	89.66	88.64	89.14
LAW	55.00	53.66	54.32	57.89	52.38	55.00	64.71	52.38	<b>57.89</b>	59.46	53.66	56.41	57.89	53.66	55.70	55.00	52.38	53.66	47.62	47.62	47.62
LAW_CASE	71.43	33.33	45.45	66.67	26.67	38.10	71.43	33.33	45.45	46.15	40.00	42.86	50.00	46.67	48.28	56.25	60.00	<b>58.06</b>	37.50	40.00	38.71
LAW_REPORT	100.00	66.67	80.00	66.67	66.67	66.67	100.00	66.67	80.00	66.67	66.67	66.67	100.00	66.67	80.00	50.00	66.67	57.14	66.67	66.67	66.67
NORP	78.57	64.71	70.97	93.33	82.35	87.50	100.00	70.59	82.76	100.00	70.59	82.76	92.86	76.47	83.87	100.00	82.35	90.32	93.75	88.24	<b>90.91</b>
ORG	64.71	67.35	66.00	78.38	59.18	67.44	73.81	63.27	68.13	73.33	67.35	70.21	78.57	67.35	72.53	80.39	82.33	<b>82.00</b>	82.93	69.39	75.56
PERSON	62.50	41.67	50.00	77.78	58.33	66.67	75.00	50.00	60.00	77.78	58.33	66.67	88.89	66.67	76.19	100.00	75.00	<b>85.71</b>	90.00	75.00	81.82
PROCEDURE	71.67	69.35	70.49	73.77	72.58	73.17	71.93	66.13	68.91	73.77	72.58	73.17	76.67	74.19	<b>75.41</b>	71.01	79.03	74.81	74.58	70.97	72.73

Table 2: Precision (P), Recall (R) and F1 score (in %) on the *cover page* and the *main text* for seven network architectures: baseline CNN model (baseline), CNN model with random static vectors on en\_core\_web\_lg (CNN+rsv), CNN with fine-tuned static vectors (CNN+fTs), CNN with random static vectors and pretraining (CNN+rsv+pt), CNN with fine-tuned static vectors and pretraining (CNN+fTs+pt), RoBERTa-based transformer, LegalBERT-based transformer

training data has a larger effect than network architecture difference. More precisely, it seems that, on some categories (CREDIBILITY, DOC\_EVIDENCE, LAW, PROCEDURE), pre-training on our own data is more effective than training on a general legal data set as in LegalBERT. This can be explained by the content LegalBERT is pre-trained on, which does not contain any Canadian but only US, European, and UK texts and does not include any refugee cases.

In other categories, roBERTa performs better than LegalBERT and CNNs, suggesting that the size of the pre-trained model is more important than domain matching. While LegalBERT has a size of 12GB, roBERTa is over 160GB and outperforms LegalBERT on traditional NER labels (GPE, ORG, PERSON and also CLAIMANT\_INFO, LAW\_CASE).

Looking at recall measures only, the superiority of transformer architectures against CNNs is more significant, with only 3 categories (DOC\_EVIDENCE, CLAIMANT\_INFO, LAW) achieving their best recall score with a CNN architecture and legal pre-training. Comparing results on recall, we reach the same conclusion as with F1, i.e. that domain matching allows us to achieve higher scores on target categories. Indeed, for seven out of 12 categories analyzed for the main text, the best scores are achieved by two architectures that differ in their pre-training domain. Higher F1 and recall scores, obtained through comparison and observation, enable us to attribute the improved performance primarily to the domain of the training data.

## 7 Related work

Because of the importance of language and written text, applications of NLP in law hold great promise in supporting legal work, which has been extensively reviewed by [Zhong et al.](#). However, because of the specificity of legal language and the diversity of legal domains, as demonstrated in our work with the results on LegalBERT-based transformer, general approaches aiming at structuring legal text such as *LexNLP* ([Bommarito II et al., 2021](#)) or general legal information extraction ([Brüninghaus and Ashley, 2001](#)) are unfit for specific domains such as international refugee law and are not able to achieve a high degree of granularity.

Earlier methods of statistical information extraction in law include the use of linear models such as maximum entropy models ([Bender et al., 2003](#); [Clark, 2003](#)) and hidden Markov models ([Mayfield et al., 2003](#)). However, state-of-the-art results are produced by methods able to capture some context, with an active research community investigating the use of conditional random fields ([Benikova et al., 2015](#); [Faruqui et al., 2010](#); [Finkel et al., 2005](#)) and BiLSTMs ([Chiu and Nichols, 2016](#); [Huang et al., 2015](#); [Lample et al., 2016](#); [Ma and Hovy, 2016](#); [Leitner et al., 2019](#)) for legal applications.

Scope and performance increased with the introduction of new architectures of deep learning using recurrent neural networks (RNN), CNNs, and attention mechanisms as demonstrated by [Chalkidis et al.](#), even though we find that transformers do not always perform best on our data. We therefore fo-



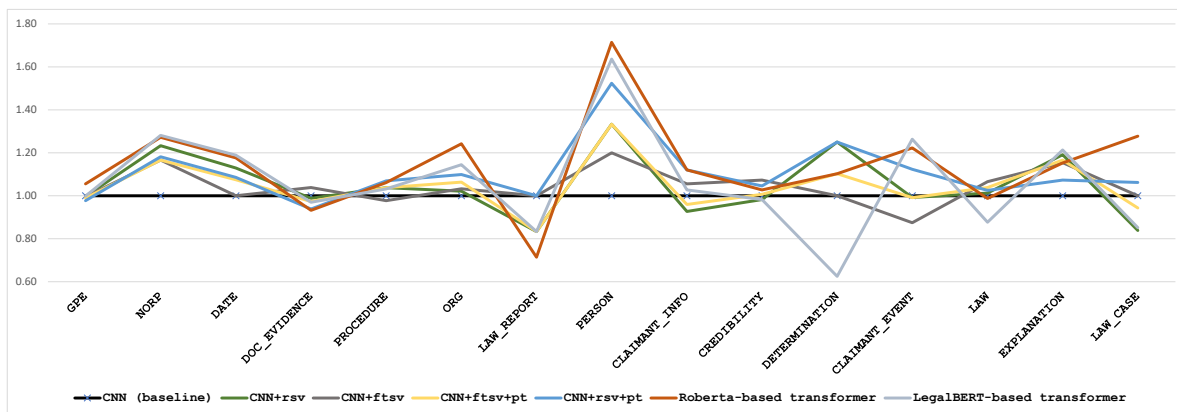


Figure 3: Comparison to the baseline on the F1 score on the *main text*: per targeted category (x-axis) on seven network architectures: baseline CNN model (baseline), CNN model with random static vectors on en\_core\_web\_lg (CNN+rsv), CNN with fine-tuned static vectors (CNN+fts), CNN with random static vectors and pre-training (CNN+rsv+pt), CNN with fine-tuned static vectors and pre-training (CNN+fts+pt), RoBERTa-based transformer, LegalBERT-based transformer

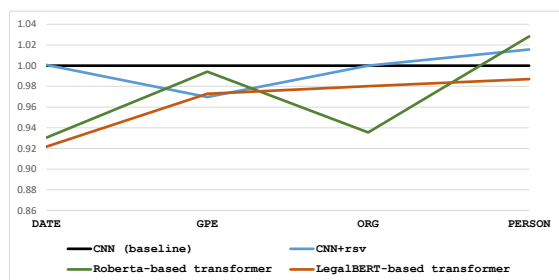


Figure 4: Comparison to the baseline on the F1 score on the *case cover*: per targeted category (x-axis) on four network architectures: baseline CNN model (baseline), CNN model with random static vectors on en\_core\_web\_lg (CNN+rsv), RoBERTa-based transformer, LegalBERT-based transformer

cus in this work on statistical NER approaches. Attempts have been made to extract legal elements of the procedure using spaCy CNNs (Pais et al., 2021; Vardhan et al., 2021) with the latter achieving a total F1 score of 59.31% across labels, citations, as well as events, which is below our reported scores.

Similar case matching is a well-known application of NLP methods, especially in common law systems (Trappey et al., 2020) and in domains such as international law. The Competition on Legal Information Extraction/Entailment includes a task of case retrieval, which proves that there is much interest in this area both from researchers and the developers of commercial applications. While research has been conducted to match cases at paragraph level (Tang and Clematide, 2021; Hu et al., 2022), we find that our approach is more transpar-

ent and shifts the decisions regarding which filters to choose to legal practitioners, which we believe is appropriate to enable productive human-machine collaboration in this high-stakes application domain.

## 8 Conclusion and future work

Our pipeline identifies and extracts diverse text spans, which may vary in quality across different categories. We acknowledge that certain entities we identify are more valuable than others for legal search purposes. Additionally, due to the complexity of the text, some noise is to be expected. However, this noise does not hinder the search for relevant items. Users have the flexibility to search and retrieve cases using any combination of our 19 categories of flagged entities. Additionally, work is required for the evaluation of the prototype by legal practitioners beyond traditional machine learning metrics (Barale, 2022). However, we believe the work presented here is an important first step and has the potential to be used for future NLP applications in refugee law. Our approach provides significant contributions with newly collected data, newly created labels for NER, and a structure given to each case based on lawyers' requirements, with nine categories of information being retrieved with an F1 score higher than 80%. Compared to existing case retrieval tools, our pipeline enables end-users to decide what to search for based on defined categories and to answer the question: *What are criteria of similarity to my new input case ?*

## Limitations

In this section, we enumerate a few limitations of our work:

- We believe that the need to train transformer architectures on GPU is an obstacle to the use of this pipeline, which is destined not to be used in an academic environment but by legal practitioners.
- Because of the specificity of each jurisdiction, generalizing to other countries may not be possible on all labels with the exact same models (for example in extracting the names of tribunals).
- The manual annotation process is a weakness: while it results in gold-standard annotations, it is very time-consuming. We do acknowledge that the amount of training data presented in this work is low and that collecting more annotations in the future would improve the quality of the results. We think it would be interesting to look at self-supervised methods, weak supervision, and annotation generation. The need for labeled data also prevents easy replication of the pipeline to new data sets, which would also require manually annotating.
- More precisely on the extracted categories, some categories lack precision and would require additional processing steps to achieve satisfactory results. For example, the category PERSON sometimes refers to the claimant or their family, but sometimes refers to the name of the judge.

## References

- Claire Barale. 2022. Human-Centered Computing in Legal NLP - An Application to Refugee Status Determination. In *Proceedings of the Second Workshop on Bridging Human-Computer Interaction and Natural Language Processing*, pages 28–33.
- Oliver Bender, Franz Josef Och, and Hermann Ney. 2003. Maximum entropy models for named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 148–151.
- Darina Benikova, Seid Muhie, Yimam Prabhakaran, and Santhanam Chris Biemann. 2015. C.: Germaner: Free open german named entity recognition tool. In *In: Proc. GSCL-2015*. Citeseer.
- Michael J Bommarito II, Daniel Martin Katz, and Eric M Detterman. 2021. Lexnlp: Natural language processing and information extraction for legal and regulatory texts. In *Research Handbook on Big Data Law*, pages 216–227. Edward Elgar Publishing.
- Stefanie Brüninghaus and Kevin D Ashley. 2001. Improving the representation of legal case texts with information extraction methods. In *Proceedings of the 8th international conference on Artificial Intelligence and Law*, pages 42–51.
- Hilary Evans Cameron, Avi Goldfarb, and Leah Morris. 2021. Artificial intelligence for a reduction of false denials in refugee claims. *Journal of Refugee Studies*, page feab054.
- Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2019. Extreme multi-label legal text classification: A case study in EU legislation. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 78–87, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.
- Daniel L. Chen and Jess Eigel. 2017. Can machine learning help predict the outcome of asylum adjudications? In *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law*, pages 237–240, London United Kingdom. ACM.
- Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the association for computational linguistics*, 4:357–370.
- Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *10th Conference of the European Chapter of the Association for Computational Linguistics*.
- Tonya Custis, Frank Schilder, Thomas Vacek, Gayle McElvain, and Hector Martinez Alonso. 2019. Westlaw edge ai features demo: Keycite overruling risk, litigation analytics, and westsearch plus. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pages 256–257.
- Nicholas Dingwall and Christopher Potts. 2018. Mitens: an extension of GloVe for learning domain-specialized representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 212–217, New Orleans, Louisiana. Association for Computational Linguistics.

- Matt Dunn, Levent Sagun, Hale Şirin, and Daniel Chen. 2017. [Early predictability of asylum court decisions](#). In *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law*, pages 233–236, London United Kingdom. ACM.
- Manaal Faruqui, Sebastian Padó, and Maschinelle Sprachverarbeitung. 2010. Training and evaluating a german named entity recognizer with semantic generalization. In *KONVENS*, pages 129–133.
- Jenny Rose Finkel, Trond Grenager, and Christopher D Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL’05)*, pages 363–370.
- Weifeng Hu, Siwen Zhao, Qiang Zhao, Hao Sun, Xifeng Hu, Rundong Guo, Yujun Li, Yan Cui, and Long Ma. 2022. Bert\_if: A similar case retrieval method based on legal facts. *Wireless Communications and Mobile Computing*, 2022.
- Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
- Kenton Lee J. Devlin, Ming-Wei Chang and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. [Neural architectures for named entity recognition](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.
- Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. 2019. Fine-grained named entity recognition in legal documents. In *Semantic Systems. The Power of AI and Knowledge Graphs*, pages 272–287, Cham. Springer International Publishing.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Xuezhe Ma and Eduard Hovy. 2016. [End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.
- James Mayfield, Paul McNamee, and Christine Piatko. 2003. Named entity recognition using hundreds of thousands of features. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 184–187.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Vasile Pais, Maria Mitrofan, Carol Luca Gasan, Vlad Coneschi, and Alexandru Ianov. 2021. [Named entity recognition in the Romanian legal domain](#). In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 9–18, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Joshua Poje. 2014. Legal research. *American Bar Association Techreport*, 2014.
- Sean Rehaag. 2007. Troubling patterns in canadian refugee adjudication. *Ottawa L. Rev.*, 39:335.
- Sean Rehaag. 2019. Judicial review of refugee determinations (ii): Revisiting the luck of the draw. *Queen’s LJ*, 45:1.
- George Sanchez. 2019. [Sentence boundary detection in legal text](#). In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 31–38, Minneapolis, Minnesota. Association for Computational Linguistics.
- Li Tang and Simon Clematide. 2021. [Searching for legal documents at paragraph level: Automating label generation and use of an extended attention mask for boosting neural models of semantic similarity](#). In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 114–122, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Charles V. Trappey, Amy J.C. Trappey, and Bo-Hung Liu. 2020. [Identify trademark legal case precedents - Using machine learning to enable semantic analysis of judgments](#). *World Patent Information*, 62:101980.
- Harsh Vardhan, Nitish Surana, and B. K. Tripathy. 2021. Named-entity recognition for legal documents. In *Advanced Machine Learning Technologies and Applications*, pages 469–479, Singapore. Springer Singapore.
- Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. [How does NLP benefit legal system: A summary of legal artificial intelligence](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5218–5230, Online. Association for Computational Linguistics.

## A Example of a case cover



RPD File No. / N° de dossier de la SPR : TA8-10977  
TA8-10773  
TA8-10812  
TA8-10930  
TA8-10931

*Private Proceeding / Huis clos*

2010 CanLII 96289 (CA IRB)

### Reasons and Decision – Motifs et décision

<b>Claimant(s)</b>	XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX (a.k.a. XXXXX XXXXX XXXXX XXXXX) XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX XXXXX	<b>Demandeur(e)s d'asile</b>
<b>Date(s) of Hearing</b>	FEBRUARY 17, 2010	<b>Date(s) de l'audience</b>
<b>Place of Hearing</b>	TORONTO, ONTARIO	<b>Lieu de l'audience</b>
<b>Date of Decision</b>	FEBRUARY 24, 2010	<b>Date de la décision</b>
<b>Panel</b>	W. LIM	<b>Tribunal</b>
<b>Counsel for the Claimant(s)</b>	STEPHEN J. SCHMIDT	<b>Conseil(s) du / de la / des demandeur(e)s d'asile</b>
<b>Tribunal Officer</b>	K. GENJAGA	<b>Agent(e) de tribunal</b>
<b>Designated Representative(s)</b>	N/A	<b>Représentant(e)s désigné(e)s</b>
<b>Counsel for the Minister</b>	N/A	<b>Conseil du ministre</b>

Figure 5: Example of a case cover

## ACL 2023 Responsible NLP Checklist

---

### A For every submission:

- A1. Did you describe the limitations of your work?  
*Not numbered, after the conclusion: 'Limitations' section*
- A2. Did you discuss any potential risks of your work?  
*Section 8 and 'Limitations'*
- A3. Do the abstract and introduction summarize the paper's main claims?  
*Abstract and section 1*
- A4. Have you used AI writing assistants when working on this paper?  
*Left blank.*

### B Did you use or create scientific artifacts?

*Section 4 and 5*

- B1. Did you cite the creators of artifacts you used?  
*Section 4 and 5 (and footnotes)*
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?  
*The tools and artifacts used are publicly available. Prodigy is the only tool used that is not freely available and we were granted an academic research license.*
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?  
*The retrieved documents are to be used for research purposes, which is the use we made of the dataset. The created artifacts (code and NER model) do not give access directly to the retrieved documents.*
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?  
*This work does not directly give access to the data but to trained models of named-entity recognition. The data is provided by the Canadian Legal Information Institute and is publicly available online.*
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?  
*Not applicable. Left blank.*
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.  
*Section 5*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

**C  Did you run computational experiments?**

*Section 6*

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?

*Section 5*

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

*Section 5*

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

*Section 6*

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

*Section 4 and 5*

**D  Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*section 5.1*

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

*After agreeing on guidelines with refugee lawyers, the first author was the only annotator involved.*

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

*Not applicable. same as above.*

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

*Not applicable. same as above.*

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

*Not applicable. same as above.*

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

*Not applicable. same as above.*