Lexata Project
Final Report

Team Members

Shubhankar Sanjay Jahagirdar - S3793593
Chandrakant Shivnath Prajapati - S3797785
Rafeed Sultaan - S3763175
Jeevitha Narayanaswamy - S3776688

Table of Contents

Table of Figures

Introduction

Lexata[1] is an intelligent security law research system which enables the lawyers to search for the specific law documents and browse through the sections of the documents and aims in reducing the efforts of browsing through the law books and redundant sections.

Leslie McCallum [2] is the CEO and the domain expert in securities law. She is the stakeholder engaging the development and implementation of the platform. She has a proposal to implement a search techniques that not only retrieves the information but also provides a summary of the top search results. As a part of WIL agreement RMIT students have collaborated to use data science techniques specifically in NLP and language processing domains to achieve the aims of Leslie's vision towards Lexata's future.

This project explains the collaborative efforts taken by the RMIT students and the Lexata stakeholders to achieve the vision of Lexata in providing the semantic searching techniques. To proceed with the requirement, an extensive research in the existing solutions and the effective difference that can add value to the existing platform was conducted. Based on this research a GPT-3 open AI platform was selected and proposed as the solution for semantic searching techniques.

GPT-3 is an auto regressive Natural Language Processing Algorithm which has been trained on 175 billion parameters. These parameters include the legal data corpus hence its abilities to recognize and process legal terms were explored and analysed. However, integration of GPT-3 with the existing platform and providing results based on the customized data corpus was a challenge.

This project follows a methodology which elaborates the complete process from collection of data to parsing and then integrating the GPT-3 engine. The GPT-3 platform is then critically analysed insights are shared with the client in a timely manner to have the feedback of the domain experts as the legal terms retrieved from the searches are legal terms. The main language engines of GPT-3 are utilized to render information on our data corpus and provide a summary of information based on the input search query. Since this is a new technology and still under development limited resources and information is published, this makes the extraction of insights a questionable process.

Overall, the project finds its scope in delivering a Minimum Viable Product which is capable of automatically parsing, dividing the information and then modelling the data. A search query functionality that is embedded provides the outputs from the GPT-3 integrated engine and summarizes the information for the end user that may or may not be domain expert. This increases the scope of Lexata as it can now target larger audience and the scope is no longer limited to the legal domain experts which is a huge gain in terms of existing market position.

Problem Statement

The problem definition as provided by the client during the initial meetings of requirement gathering was to develop a product that can produce semantic search results. This effectively entails the procedure that reduces the human efforts required in collection, parsing, and storing the law documents.

Approach of the project

The project follows an iterative development approach in which client meetings and feedbacks on the deliverables were a part of the development process. Weekly meetings were conducted where user stories and requirements were gathered and solutions to those requirements were presented on an iterative basis.

The main aim of this project was to automate the process of semantic search using the Open AI's GPT-3 engine. This project focuses on end-to-end creation of solution which includes steps from the data extraction from various sources to parsing the legal documents from documents in the form of PDF. Further this parsed data is used to train the GPT-3 model later this GPT-3 can be integrated with web framework to mimic the functionalities of existing Lexata website.

User story and requirements

- The client presented the team with the Law documents in PDF and website links format.
- The existing Lexata platform was presented which searches the documents based on the Lexical search paradigms and produces results having an exact match to the query.
- The requirement of the client specified the use of Open AI platform and GPT-3 engine to model the information and present the semantic search results.
- The granular level requirements were specifically integrated in the product on an interactive basis and the data pipeline was developed as described in the figure below.
- A Most Viable Product (MVP) was a requirement which renders the search results of a user queries based on the GPT-3 search scores and produces results on the selected engines as can be seen in Figure 1.

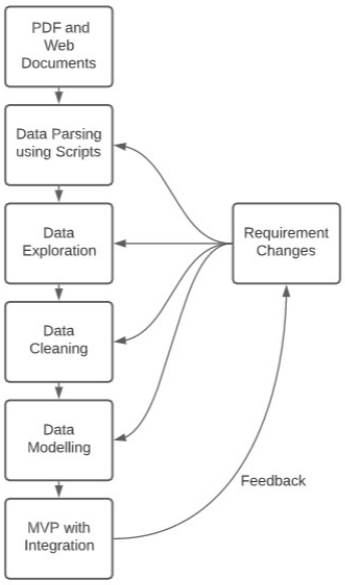


Figure 1: Data pipeline

Methodology

This section explains the thorough process of collecting parsing and scripting of the 3 main data sources, Ontario Exemptive Relief, British Columbia Securities Law, Ontario Specific Securities law. Once the collection and parsing are done, we explain the Data modelling techniques which includes integration of GPT-3 Open AI. Then the evaluation and comparative analysis is performed.

Ontario Exemptive Relief Dataset

The data for the Ontario Exemptive Relief was extracted from the Ontario Securities Commission website [3]. A snippet of the Ontario Securities Commission website's search engine with keyword "exemptive relief" is shown in Figure 2. The search query generates 433 pages with 20 search results in each page. A total of 8699 documents were returned as result of the search query. Each page contains a table structure with 5 columns namely Date, Title, Instrument, Document Type and Related-To. These columns form the meta-tags for the document that we wish to parse.

If we click on the values inside the "Title" column, it redirects to the URL of the actual document that we need to parse. The structure of the document is shown in Figure 3 and Figure 4, where the document is divided into 3 parts namely Headnote, Applicable Legislations and Content.

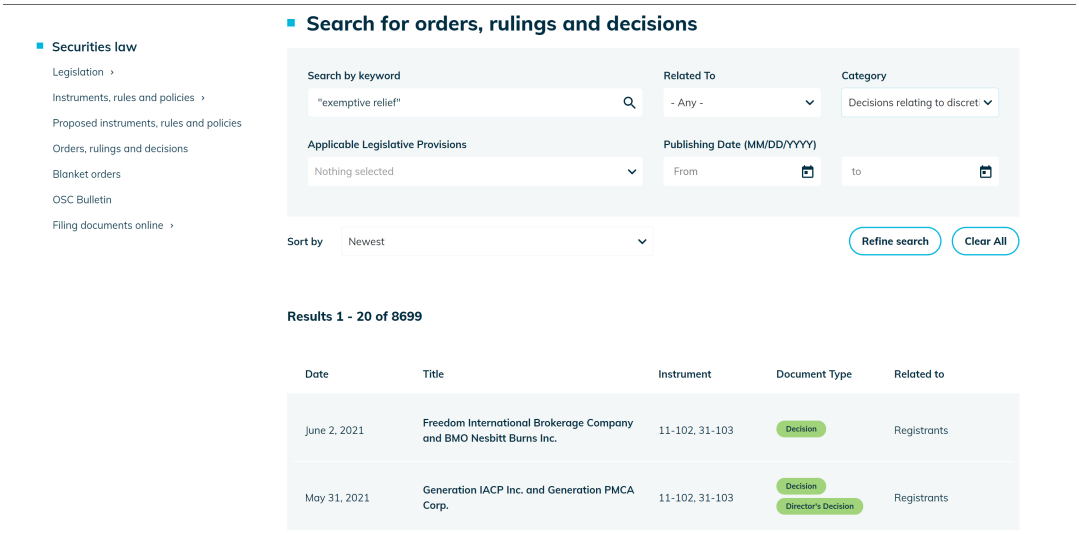


Figure 2: Structure of the search engine on the website with keyword "exemptive relief" on the website.

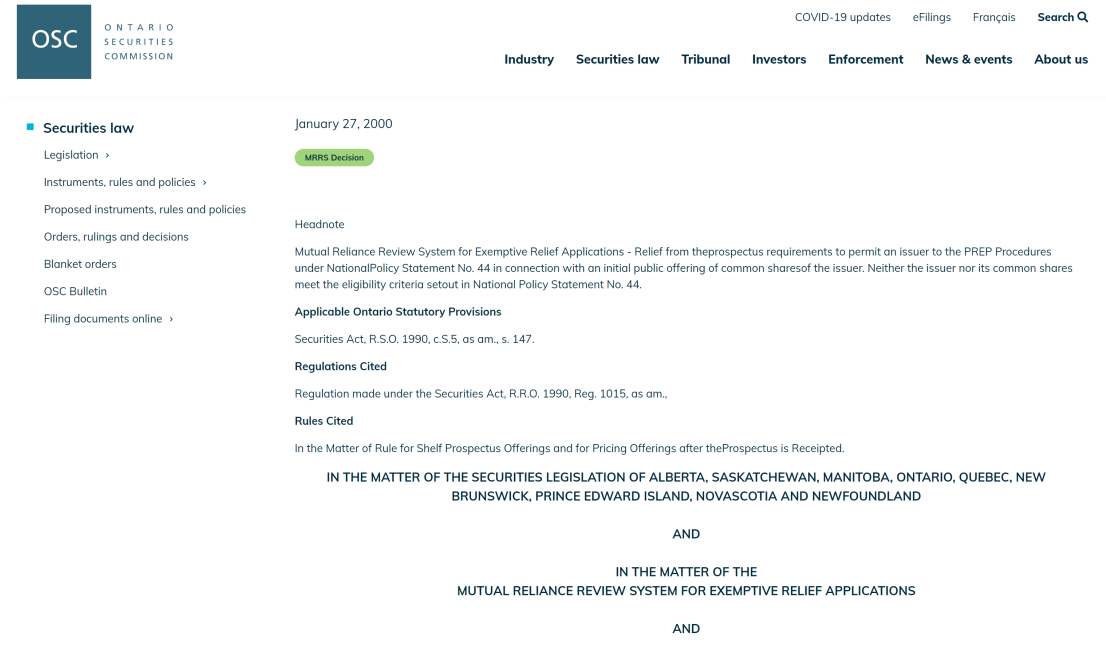
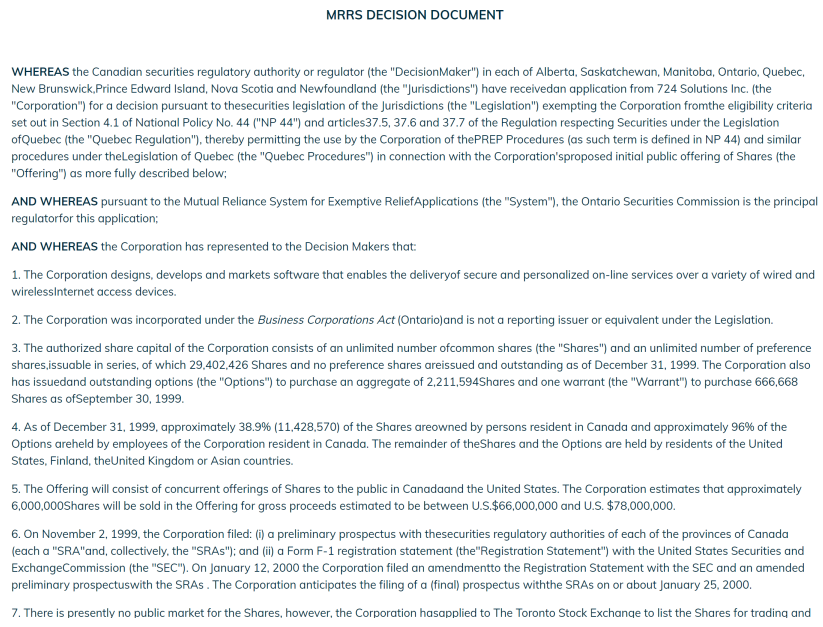


Figure 3: Structure of a sample document of Ontario Exemptive Relief highlighting Headnote and Applicable Legislation on the Website



MRRS DECISION DOCUMENT

WHEREAS the Canadian securities regulatory authority or regulator (the "DecisionMaker") in each of Alberta, Saskatchewan, Manitoba, Ontario, Quebec, New Brunswick,Prince Edward Island, Nova Scotia and Newfoundland (the "Jurisdictions") have receivedan application from 724 Solutions Inc. (the "Corporation") for a decision pursuant to thesecurities legislation of the Jurisdictions (the "Legislation") exempting the Corporation fromthe eligibility criteria set out in Section 4.1 of National Policy No. 44 ("NP 44") and articles37.5, 37.6 and 37.7 of the Regulation respecting Securities under the Legislation ofQuebec (the "Quebec Regulation"), thereby permitting the use by the Corporation of thePREP Procedures (as such term is defined in NP 44) and similar procedures under theLegislation of Quebec (the "Quebec Procedures") in connection with the Corporation'sproposed initial public offering of Shares (the "Offering") as more fully described below;

AND WHEREAS pursuant to the Mutual Reliance System for Exemptive ReliefApplications (the "System"), the Ontario Securities Commission is the principal regulatorfor this application;

AND WHEREAS the Corporation has represented to the Decision Makers that:

1. The Corporation designs, develops and markets software that enables the deliveryof secure and personalized on-line services over a variety of wired and wirelessInternet access devices.

2. The Corporation was incorporated under the *Business Corporations Act* (Ontario)and is not a reporting issuer or equivalent under the Legislation.

3. The authorized share capital of the Corporation consists of an unlimited number ofcommon shares (the "Shares") and an unlimited number of preference shares,issuable in series, of which 29,402,426 Shares and no preference shares areissued and outstanding as of December 31, 1999. The Corporation also has issuedand outstanding options (the "Options") to purchase an aggregate of 2,211,594Shares and one warrant (the "Warrant") to purchase 666,668 Shares as ofSeptember 30, 1999.

4. As of December 31, 1999, approximately 38.9% (11,428,570) of the Shares areowned by persons resident in Canada and approximately 96% of the Options areheld by employees of the Corporation resident in Canada. The remainder of theShares and the Options are held by residents of the United States, Finland, theUnited Kingdom or Asian countries.

5. The Offering will consist of concurrent offerings of Shares to the public in Canadaand the United States. The Corporation estimates that approximately 6,000,000Shares will be sold in the Offering for gross proceeds estimated to be between U.S.$66,000,000 and U.S. $78,000,000.

6. On November 2, 1999, the Corporation filed: (i) a preliminary prospectus with thesecurities regulatory authorities of each of the provinces of Canada (each a "SRA"and, collectively, the "SRAs"); and (ii) a Form F-1 registration statement (the"Registration Statement") with the United States Securities and ExchangeCommission (the "SEC"). On January 12, 2000 the Corporation filed an amendmentto the Registration Statement with the SEC and an amended preliminary prospectuswith the SRAs . The Corporation anticipates the filing of a (final) prospectus withthe SRAs on or about January 25, 2000.

7. There is presently no public market for the Shares, however, the Corporation hasapplied to The Toronto Stock Exchange to list the Shares for trading and

Figure 4: Structure of a sample document of Ontario Exemptive Relief highlighting content on the website

Parsing Ontario Exemptive Relief Data

All the data was stored online and now we needed to perform web-scraping to pull the data from the web into a tabular structure with certain rows and columns. We used a python script with BeautifulSoup Library to scrap the meta-tags and the content. We first scraped the meta-tags from the Ontario Securities Commission website [3] scraping the meta-tags from the table structure within this webpage and repeated the same process for all the 433 pages using a loop. The contents and the URLS of the actual document are extracted from HTML DOM elements. Next, we redirected to the actual documents. These documents contained “Headnote” and “Applicable Legislation” and “Content” contained inside HTML tags “<p>”. By studying the pattern inside the documents, we were able to extract the meta-tags and the content of the document. Figure 5 shows a snippet of the python script which can be found in Lexata’s GitHub Repository.

Generating meta tags for OSC files names, url with meta-tags and content

In [13]:
```
#Using the static url with relative addressing
url_prefix = 'https://www.osc.ca/en/securities-law/orders-rulings-decisions?keyword=%22exemptive%20relief%22&field_ord_related_to=All&fie
ld_ord_category=5516&date%5Bmin%5D=&date%5Bmax%5D=&sort_bef_combine=field_publication_date_DESC&sort_by=field_publication_date&sort_order
=DESC&page='

#the number of pages is 433, indexing starts from 0
page_start = 0
page_end = 433

# Storing each category as  a list to be converted to dataframe later
#i.e. the meta tags

date = []
title =[]
instrument = []
document_type =[]
related_to = []
urls = []
documents_with_formatting = []
contents = []
headnotes = []
applicable_legistlations = []

print("Generating meta tags for OSC files names, meta-tags and content")


#Processing the content of each page to generate the meta tags, file name, url,content and the
#the whole document preserving formatting

for page_index in range(page_start,page_end):
    print("Processing page number "+str(page_index+1)+" ...") #comment

    #Generating new urls for each page number
    url = url_prefix + str(page_index)

    #Parsing the html file using Beautiful Soup Library
    response = requests.get(url)
    soup = BeautifulSoup(response.text, "html.parser")

    #Extracting the table contents using html tag table-listings__content
    block = soup.find(attrs={'class':"table-listings__content"})



    # Scrapping the contents one by one and storing them as link.
    content_list = []
    for child in block.children:
        content_list.append(child)
        #print(child)

    # Storing each row values.
    Data = []
    for i in range(len(content_list)):
        #Unnecessary new lines are ignored and white space trimmed
        if content_list[i] != '\n':
            subcon = []
            for string in content_list[i].stripped_strings:
                subcon.append(repr(string))
            Data.append(subcon)
        else:
            continue
```

Figure 5: Snippet of the Python script which can be found here

Ontario Exemptive Relief Data Description

The Ontario Securities Commission website [3] is constantly getting updated and the search query “exemptive relief” on April 15th returned resulted in 8027 results. As a result, the dataset “osc_files_data.csv” contains 8027 rows and 8 columns. Figure 5 shows a snippet of the Ontario Exemptive Relief dataset. The columns with their respective descriptions are:

1) Date: The “Date” variable contains the published date of the document

2) Title: The “Title” variable contains the title of the document.

3) Instrument”: The “Instrument” variable contains the instrument number. Each document contains a unique instrument number.

4) RelatedTo: The “RelatedTo” variable contains information about what other documents are related to this document.

5) URL: The “URL” variable contains information about where the document is on the web.

6) Headnote: The “Headnote” variable contains information about

7) ApplicableLegislation: The “Applicable” Legislation contains information about the legislations that apply to this document.

8) Content: The “Content” variable contains the actual content in the document. The content variable does not include the “Headnote” and “ApplicableLegistlation.”

	Date	Title	Instrument	DocumentType	RelatedTo	URL	Headnote	ApplicableLegistlation	Content
0	March 29, 2021	T. Rowe Price (Canada), Inc. and T. Rowe Price...	81-106	Decision	Investment funds and structured products	https://www.osc.ca/en/securities-law/orders-ru...	National Instrument 11-203 Process for Exempti...	National Instrument 81-106 Investment Fund Con...	Applicable Legislative ProvisionsNational Inst...
1	March 26, 2021	Vanguard Investments Canada Inc.	Securities Act	Decision	Investment funds and structured products	https://www.osc.ca/en/securities-law/orders-ru...	National Policy 11-203 Process for Exemptive R...	Securities Act, R.S.O. 1990, c. S.5, as am., s...	Applicable Legislative ProvisionsSecurities Ac...
2	March 25, 2021	AngelList Holdings, LLC and AngelList Advisors...	31-103, Securities Act	Decision	Registrants	https://www.osc.ca/en/securities-law/orders-ru...	CSA Regulatory Sandbox initiative -- Prior dec...	Statutes Cited	Applicable Legislative ProvisionsStatutes Cite...
3	March 19, 2021	RP Investment Advisors LP	81-101	Decision	Investment funds and structured products	https://www.osc.ca/en/securities-law/orders-ru...	National Policy 11-203 Process for Exemptive R...	National Instrument 81-101 Mutual Fund Prospec...	Applicable Legislative ProvisionsNational Inst...
4	March 19, 2021	Lysander Funds Limited	11-203, 81-101	Decision	Investment funds and structured products	https://www.osc.ca/en/securities-law/orders-ru...	National Policy 11-203 Process for Exemptive R...	National Instrument 81-101 Mutual Fund Prospec...	Applicable Legislative ProvisionsNational Inst...
5	March 19, 2021	CI Investments Inc	Decision	Investment funds and structured products	Investment funds and structured products	https://www.osc.ca/en/securities-law/orders-ru...	NP 11-203 Process for Exemptive Relief Applica...	National Instrument 81-102 Investment Funds, s...	Applicable Legislative ProvisionsNational Inst...

Figure 6: Ontario Exemptive Relief Dataset

British Columbia Securities Law dataset

The data collected is collected from the website, British Columbia Securities Commission [4]

The data was collected from the following sections according to Lexata's Specification from following sections:

- Procedure & Related Matters
- Certain Capital Market Participants
- Registration Requirements & Related Matters
- Distribution Requirements
- Ongoing Requirements for Issuers & Insiders
- Takeover Bids & Special Transactions
- Securities Transactions Outside the Jurisdiction
- Investment Funds
- Derivatives

We have downloaded all the files according to the client's specification and stored them in a hierarchical file structure, where the data is divided into 9 sections. Each of these sections contains folders specifying the "document code" and "document name". This folder further contains documents like Companion Policy, National Instrument, British Columbia Instrument, Multinational Instrument, CSA Notice and other documents. There is a separate folder inside the document folders to contain "FORMS" to contain the different type of forms as specified by Lexata, Inc. We have downloaded roughly 500 pdf documents and stored them in a hierarchical structure, since we could not find a way to automate the pdf downloading process. The website has a problem to stop the mass downloading of files if we use a script or a separate software called "UIPath" to download all these files. Therefore, this process was extremely time consuming.

Only the Current documents were downloaded. The historical and proposed documents were skipped as directed by Lexata, Inc. According to Lexata's specification below, we have chosen the sections as specified the figure below.

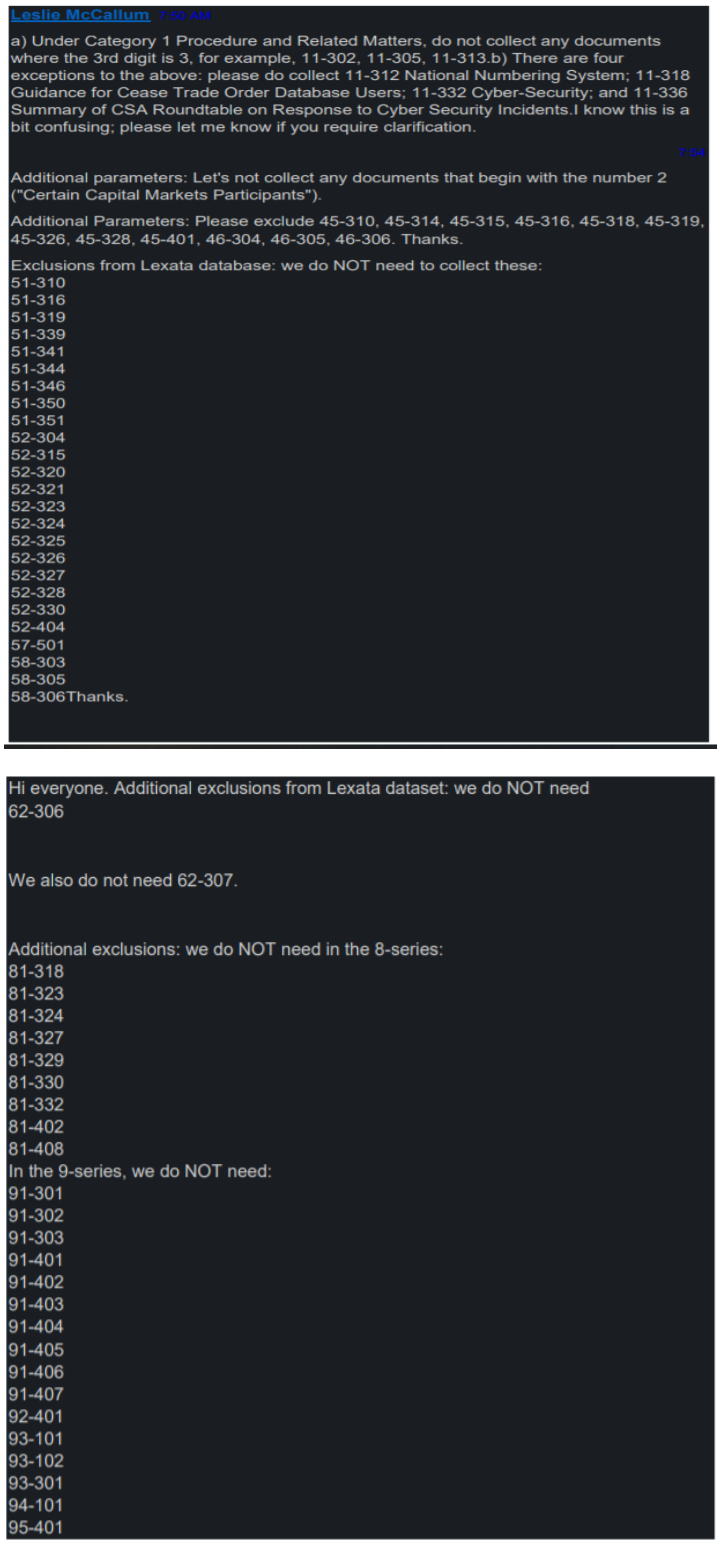


Figure 7: Data Collection Requirements

Each of these sections contains multiple documents in pdf format, for example, 11-102 Passport System till 15-904 Endorsement of Warrant. Each of these sections contain around 60 documents. With some of these documents contains as high as 90 pages each. The average length of these documents average around 40 pages.

The formatting of these documents is variable that is there is no fixed formatting in these pdf documents.

The document types include:

- BC Instrument/BC policy
- National Instrument/ National Policy
- Multi-National Instrument
- Companion Policy
- CSA Notice
- Forms

Other documents for the parsing phase, we are going to be ignoring the FORMS as directed by the requirements of the client. These documents are being stored in 6 main meta tags namely Document Type, Document Code, Document Name, Part, Section and Content.

Challenges and issues with British Columbia Securities Law Documents

As you can see from Figure 10, the document head contains the document type (i.e., Companion Policy, 11-102CP), the document name (i.e. Passport System), the document code (i.e. 11-102CP), Part (i.e. Part-1 General) and Section (1.1 Definitions). The page number (i.e., 3 is a redundant value which will be taken care of by the data cleaning process). The rest of the text is the content. Notice in Figure 8, the Section names are missing on the document to the right, and the title has extra added information "to-National Instrument 31-102". The case sensitivity of the "part number" is altered as we progress through the document.

The documents are inconsistent in terms of font size, styles are present. The number of variations increases as we look over large volumes of documents. Moreover, we can see that CSA Staff notice contains a banner in Figure 12. We cannot extract text from the document except for using Image Recognition Software to extract title. As of now, we will not be able to store table and images present in these documents in CSV or JSON format, which are the general formats to query the recommendation system using GPT-3. The table structure cannot be preserved in CSV format or JSON Format. Moreover, we have chosen to ignore the Appendix of these documents because most of these appendices contains tables and images. Figure 9 shows the example of irrelevant information stored inside these documents that needs to be discarded.

Companion Policy 11-102CP
Passport System

PART 1 GENERAL

1.1 Definitions

In this Policy,

"CP 33-109" means Companion Policy 33-109CP *Registration Information*;

"domestic firm" means a firm whose head office is in Canada;

"domestic individual" means an individual whose working office is in Canada;

"MI 11-101" means Multilateral Instrument 11-101 *Principal Regulator System*;

"non-principal jurisdiction" means, for a person or company, a jurisdiction other than the principal jurisdiction;

"non-principal regulator" means, for a person or company, the securities regulatory authority or regulator of a jurisdiction other than the principal jurisdiction;

"NP 11-202" means National Policy 11-202 *Process for Prospectus Reviews in Multiple Jurisdictions*;

"NP 11-203" means National Policy 11-203 *Process for Exemptive Relief Applications in Multiple Jurisdictions*;

"NP 11-204" means National Policy 11-204 *Process for Registration in Multiple Jurisdictions*;

"NP 11-205" means National Policy 11-205 *Process for Designation of Credit Rating Organizations in Multiple Jurisdictions*;

"NP 11-206" means National Policy 11-206 *Process for Cease to be a Reporting Issuer Applications*;

"NRD" has the same meaning as in NI 31-102;

"NRD format" has the same meaning as in NI 31-102;

"SRO" means a self-regulatory organization; and

"T&C" means a term, condition, restriction or requirement imposed by a securities regulatory authority or regulator on the registration of a firm or an individual.

Companion Policy 31-102CP
to National Instrument 31-102 *National Registration Database*

Part 1 Purpose

The purpose of NI 31-102 is to establish requirements for the electronic submission of registration information through NRD. References in this policy to "we" mean the securities regulatory authority and regulator.

Part 2 Production of NRD Filings

The securities legislation of several jurisdictions contains a requirement to produce or make available an original or certified copy of information filed under the securities legislation. We consider that it may satisfy such a requirement in the case of information filed in NRD format by providing a printed copy or other output of the information in readable form that contains or is accompanied by a certification by the securities regulatory authority or regulator that the printed copy or output is a copy of the information filed in NRD format.

Part 3 Date of Filing

We think that information filed in NRD format is, for purposes of securities legislation, filed on the day that the transmission of the information to NRD is completed.

Part 4 Official Copy of NRD Filings

For purposes of securities legislation, securities directions or any other related purpose, we think that the official record of any information filed in NRD format by an NRD filer is the electronic information stored in NRD.

Part 5 Authorized Firm Representative as Agent

We think that when making an NRD submission an AFR is an agent of the firm or individual to whom the filing relates.

Part 6 Ongoing Firm Filer Requirements

We expect that firm filers will follow the processes set out in the NRD User Guide to:

(a) enrol with the NRD administrator;

(b) keep their enrolment information current; and

(c) keep their NRD account information current.

CONDITIONAL EXEMPTION FROM REGISTRATION FOR

UNITED STATES BROKER-DEALERS AND AGENTS

PART 1 INTRODUCTION

1.1 Introduction - Cross-border trading activities between Canada and the United States of America often take place because of the movement of residents between the two countries. In order to facilitate certain cross-border trading activities that may arise between United States broker-dealers and their existing clients who are now located in Canada, the Canadian securities regulatory authorities have adopted National Instrument 35-101 Conditional Exemption From Registration for United States Broker-Dealers and Agents (the "Instrument") which provides certain broker-dealers, and their agents, resident in the United States of America with a conditional exemption from the applicable registration requirements and the prospectus requirement. This approach is consistent with the Instrument's underlying policy that investors will be relying primarily upon the regulation by securities regulators and statutory liability imposed by legislation in the broker-dealer's or agent's home jurisdiction for protection.

PART 2 GENERAL PRINCIPLES

2.1 General - The Instrument provides that a United States broker-dealer and its agents may engage in two specific types of cross-border trading activities in foreign securities with an individual who was previously resident in the United States of America, and is now located in Canada, regardless of nationality. In Quebec, the term foreign securities includes futures.

2.2 Temporarily Resident - The first category of activity provided for under clause 2.1(c)(i) and clause 3.1(d)(i) of the Instrument permits brokers-dealers and their agents to deal in foreign securities with an individual ordinarily resident in the United States of America who is temporarily resident in a Canadian jurisdiction and with whom the broker-dealer had a broker-dealer client relationship before the individual became temporarily resident in the Canadian jurisdiction. This aspect of the Instrument is intended to allow persons from the United States who are on a temporary work assignment in Canada, or who may be in Canada on vacation or for other reasons, to trade with their home broker-dealer and agent in the United States of America. The concept of "temporarily" as it appears in the National Instrument is based upon SEC Rule 15a-6 which exempts certain non-United States broker-dealers from registering under the 1934 Act.

The Canadian Securities Administrators are of the view that a person that ceases to be "ordinarily resident" in the United States of America would not retain status as a United States resident "temporarily resident" in Canada under the Instrument.

2.3 Tax-Advantaged Plans - The second category of activity provided for under clause 2.1(c)(ii) and clause 3.1(d)(ii) of the Instrument permits broker-dealers and their agents to deal in foreign securities with an individual who was previously resident in the United States of America and who is resident in a Canadian jurisdiction for trades for and with the individual's tax-advantaged retirement savings plan (for example, an Individual Retirement Account), if the plan is located in the United States and the individual is either a holder of, or contributor to, the plan. Under laws of the United States of America, tax-advantaged retirement savings plans must be located in the United States of America and result in adverse tax consequences for United States individuals if collapsed. For these reasons, individuals are permitted by the Instrument to continue this type of trading activity with a broker-dealer and its agent in the United States of America whether or not there was a pre-existing relationship with the broker-dealer or agent while the individual was in the United States of America.

Figure 8: Variation of structure and format in documents of the same type

Questions
If you have questions about this Notice please direct them to any of the following:

Michael Brady
Senior Legal Counsel, Capital Markets Regulation
British Columbia Securities Commission
Tel: 604-899-6561
1-800-373-6393
mbrady@bcsc.bc.ca

4

Navdeep Gill
Legal Counsel, Market Regulation
Alberta Securities Commission
Tel: 403-355-9043
navdeep.gill@asc.ca

Curtis Brezinski
Compliance Auditor
Saskatchewan Financial Services Commission
Tel: 306-787-5876
curtis.brezinski@gov.sk.ca

Chris Besko
Legal Counsel, Deputy Director
The Manitoba Securities Commission
Tel. 204-945-2561
Toll Free (Manitoba only) 1-800-655-5244
chris.besko@gov.mb.ca

Christopher Jepson
Senior Legal Counsel
Compliance and Registrant Regulation
Ontario Securities Commission
Tel: 416-593-2379
cjepson@osc.gov.on.ca

Sophie Jean
Conseillère en réglementation
Surintendance de l'assistance à la clientèle, de l'indemnisation et de la distribution
Autorité des marchés financiers
Tel: 514-395-0337, ext. 4786
Toll-free: 1-877-525-0337
sophie.jean@lautorite.qc.ca

Brian W. Murphy
Deputy Director, Capital Markets
Nova Scotia Securities Commission
Tel: 902-424-4592
murphybw@gov.ns.ca

FREQUENTLY ASKED QUESTIONS

	QUESTION	ANSWER
General Questions		
1.	When does someone cease to be a client, such that a registrant is no longer required to provide the statements and reports contemplated in the CRM2 Amendments?	It is not possible to provide a bright line test for determining when a client relationship has ended that will apply in all cases. We expect firms to exercise reasonable professional judgement, erring in favour of providing client reporting where there is doubt as to whether there is still a client relationship. Some principles that apply to the exercise of that judgement are: • A person remains a client of a registered dealer or adviser for so long as the dealer or adviser holds securities owned by the person, or the circumstances described in subsection 14.14.1(1) [*additional statements*] apply. • A firm should consider the totality of its dealings with a client and the client's expectations of ongoing services from the firm. • Whether a client relationship is ongoing or not depends on the particular facts and circumstances of the relationship. Note that a registered dealer or adviser may not avoid the client reporting requirements in NI 31-103 by selectively choosing to cease to be the dealer of record for some of a client's securities. For example, a dealer may not tell the IFM of a client's mutual funds that it is no longer the dealer of record for some of the client's securities (unless those securities have been transferred to an account of the client at another dealer or an adviser), and at the same time, keep an account for the client. See also the guidance in question 35 regarding section 14.15 [*security holder statements*].
2.	Do disclosure and reporting requirements in CRM2 Amendments apply to other investments that may not be securities, such as segregated funds?	The jurisdiction of the CSA limits the CRM2 Amendments to securities (including derivatives or exchange contracts, as applicable, in certain jurisdictions pursuant to the requirements of section 1.2 of NI 31-103). Nonetheless, we encourage registrants to provide their clients with information that meets the standards set in the CRM2 Amendments in respect of all of their investments. This will enable investors to better understand the relative costs of different investments and their performance. Note that requirements imposed by SROs may extend to such investments.
3.	Where should switch fees and short-term trading fees be reported?	Switch fees charged by a registered dealer or adviser are considered a transaction charge (see the discussion of the definition of "transaction charge" in section 14.2 of the CP). They must be disclosed before the trade (section 14.2.1), in a trade confirmation (paragraph14.12(1)(c)) and in the annual report on charges and other compensation (paragraph 14.17(1)(c)). Short-term trading fees paid to an investment fund must be disclosed in a trade confirmation but are not included in the requirements for the annual report on charges and other compensation.
14.2	**Relationship disclosure information**	

DM#1821571 Page 5 of 16

British Columbia Securities Commission

BC Instrument 33-513

The British Columbia Securities Commission, considering that to do so would not be prejudicial to the public interest, orders that effective September 28, 2009, BC Instrument 33-513 entitled *Exemption from financial statement, capital and bonding requirements for MFDA Members*, dated November 25, 2003, is revoked and the attached BC Instrument 33-513 *Exemption from capital and bonding requirements for MFDA Members* is made.

September 21, 2009

Brent W. Aitken
Acting Chair

(This part for administrative purposes only and is not part of the Order)

Authority under which Order is made:
Act and sections:- *Securities Act*, sections 171 and 48(1)
Other (specify):-

Figure 9: Examples of irrelevant information stored in BC documents

Automated Parsing

We have found three types of document templates in the British Columbia Securities Document as seen in Figure 10, Figure 11 and Figure 12. These document templates form the basis of the automated scripts for parsing the pdf documents.

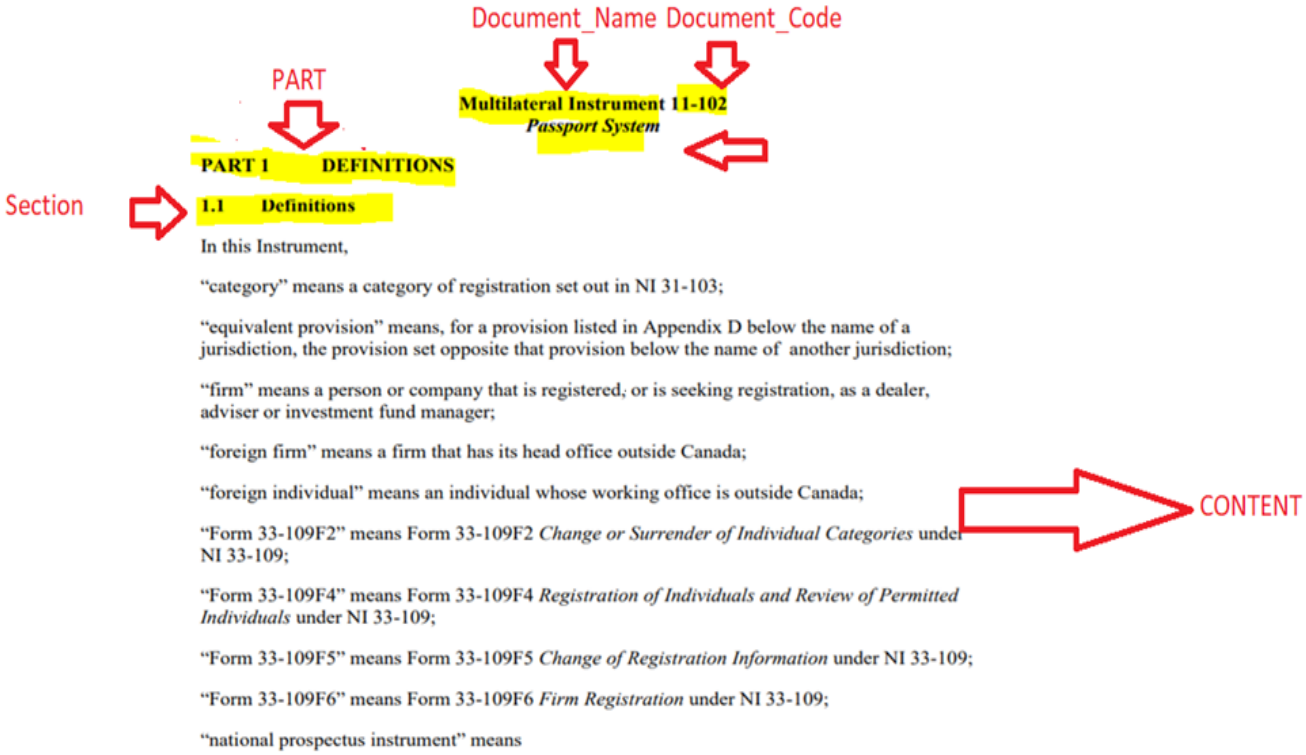


Figure 10:Multilateral Instrument, National Instrument, National Policy, Companion Policy Template

British Columbia Securities Commission

Document Type ⇦ **BC Instrument 33-519** ⇨ Document Code

Exemption for Investment Industry Regulatory Organization of Canada Members from Certain Requirements of National Instrument 31-103 Registration Requirements, Exemptions and Ongoing Registrant Obligations ⇨ Document Name

Interpretation

Terms defined in the *Securities Act*, National Instrument 31-103 *Registration Requirements, Exemptions and Ongoing Registrant Obligations* (**NI 31-103**) and National Instrument 14-101 *Definitions* have the same meaning in this Instrument.

Part ⇦ **Background**

Section ⇦ 1. Under section 9.3 [*exemptions from certain requirements for IIROC members*] of NI 31-103, a registered firm that is a member of IIROC is exempt from certain requirements in NI 31-103 if the registered firm complies with the corresponding IIROC Provisions that are in effect. The term "IIROC Provision" is defined in section 1.1 of NI 31-103 to mean "a by-law, rule, regulation or policy of IIROC named in Appendix G, as amended from time to time". ⇨ Content

2. On July 15, 2014, the following provisions of NI 31-103 will come into effect:

(a) paragraph 14.2(2)(m) [*relationship disclosure information*];

Figure 11: British Columbia Instrument and British Columbia Policy Template

CSA/ACVM Canadian Securities Administrators / Autorités canadiennes en valeurs mobilières

CSA Staff Notice 31-345

Cost Disclosure, Performance Reporting and Client Statements – Frequently Asked Questions and Additional Guidance

⇨ This is an image that contains Document code, Document Name and Document Type

Date ⇦ **April 14, 2016**

Background

Amendments to National Instrument 31-103 *Registration Requirements, Exemptions and Ongoing Registrant Obligations* (**NI 31-103**) and Companion Policy 31-103CP *Registration Requirements, Exemptions and Ongoing Registrant Obligations* (**31-103CP** or the **CP**) implementing phase 2 of the Client Relationship Model (**CRM2**) came into force on July 15, 2013 (the **CRM2 Amendments**). Staff of the Canadian Securities Administrators (**CSA staff** or **we**) have compiled these frequently asked questions and our responses as well as further guidance (**FAQs**) in addition to that which we published in CSA Staff Notice 31-337 *Cost Disclosure, Performance Reporting and Client Statements – Frequently Asked Questions and Additional Guidance as of February 27, 2014* (**CSA SN 31-337**). FAQs from CSA SN 31-337 have been consolidated with the further FAQs in this notice. For that reason, CSA SN 31-337 is hereby withdrawn. Some of the earlier FAQs have been superseded in part by the further FAQs or left out of this consolidation because they are no longer necessary. Among other things, this notice includes a section on the applicability of the CRM2 Amendments to exempt market dealers. Some parts of this guidance were previously published in CSA Staff Notice 31-324 *Exempt Market Dealers and Account Statement Requirements in National Instrument 31-103 Registration Requirements and Exemptions* dated June 22, 2011 (**CSA SN 31-324**). With the publication of the updated guidance in this notice, CSA SN 31-324 is also hereby withdrawn. ⇨ Content

In this notice, "**registered firm**" or "**firm**" includes both registered dealers and registered advisers unless otherwise specified, and we refer to mutual fund dealers as "**MFDs**", exempt market dealers as "**EMDs**", portfolio managers as "**PMs**" and investment fund managers as "**IFMs**".

All references in this notice to sections, subsections, paragraphs and subparagraphs are to NI 31-103, unless otherwise noted.

Figure 12: British Columbia CSA Staff Notice Template

The algorithm for the automated scripts is highlighted in Figure 13. In summary, we identify the patterns within the document to see if it falls under any one of the document templates shown in Figure 10,11,12. After we manually examine the pages to determine the irrelevant pages within the document, we trim them from the data. Next, we tokenize the documents using Regular Expression. In those tokens, we look to find our predefined meta-tags like document type, document code, document name, date and parts. Our scripts are modified manually for variations like changes in font and cases. After that, we look for unique token names, for example "Part", and extract the appropriate contents from the documents using indexes found by utilizing a loop structure. Finally, we generate the data frame that is merged with our final dataset. The results of the algorithm can be shown in the Figure 14,15.

As a result, we have created three scripts which can handle the following document types of BC Instrument: BC Policy, National Instrument, National Policy, Companion Policy, Multi-lateral Instrument, Rules and CSA Staff Notice.

The British Columbia Policy and British Instrument are structured documents, where information such as Document Name, Document Type, Document Code and Order can be extracted from the documents using simple python scripting using PYPDF2 library in python 3. The script used for this task is called "Automated_Templated_Script_For_Parsing_BC_Policy_BC_Instrument.ipynb". The documents are structured into 4 parts: Background/Definitions, Interpretation, Order and Effective Date. Using regular expression, we were able to extract the contents of each of these sections. As the sections number did not have a distinct token name, we were not able perform section label parsing. There is no identifier in these documents to determine sub-parts or sub-sections inside these documents.

The National Instrument, National Policy, British Columbia Policy, British Columbia Instrument, Result-lateral Instrument and Companion Policy follow the same structure of Document Type, Document Name and Document Code. Each content of the document is divided into parts and sections. But the main limitation of these documents is that there is no unique "token name" for section. As a result, we were able to perform part-level parsing because "PART" is the only unique token name that appears inside these documents. We have extracted content for each unique part inside the content column.

The CSA-Staff Notice had no structure in the documents except date and the meta-tags like Document Type, Document Code and Document Code are hidden inside the banner which is an image. There is a need for OCR or human-level interaction to extract that information. We were able to extract the date and the content of the documents. The "content" column is cleaned to contain only relevant information. Content after "For more Information:" token is removed since the content would then include unnecessary information like email address and appendix.

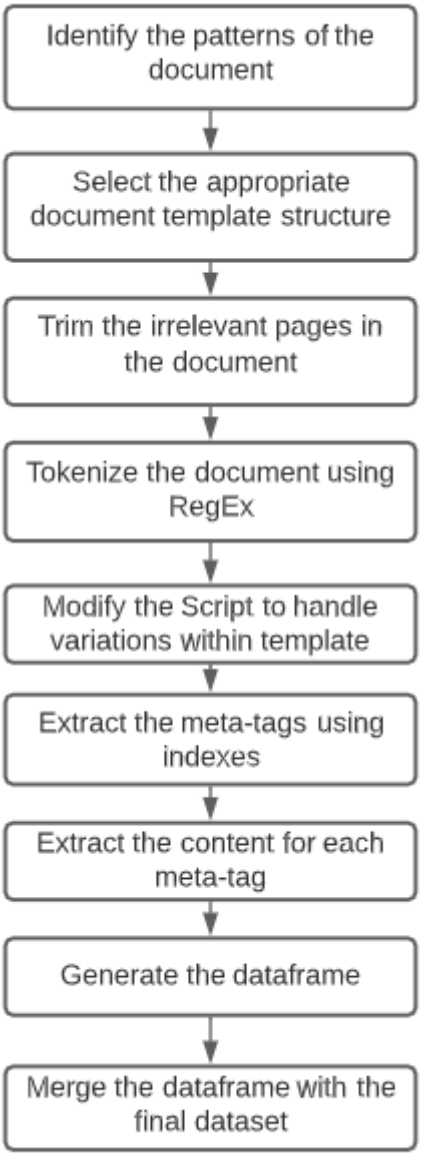


Figure 13: Algorithm used for parsing the BC securities documents

ORIGNAL DOCUMENT

PART 1 DEFINITIONS

1.1 Definitions

In this Instrument,

"category" means a category of registration set out in NI 31-103;

"equivalent provision" means, for a provision listed in Appendix D below the name of a jurisdiction, the provision set opposite that provision below the name of another jurisdiction;

"firm" means a person or company that is registered, or is seeking registration, as a dealer, adviser or investment fund manager;

"foreign firm" means a firm that has its head office outside Canada;

"foreign individual" means an individual whose working office is outside Canada;

"Form 33-109F2" means Form 33-109F2 *Change or Surrender of Individual Categories* under NI 33-109;

"Form 33-109F4" means Form 33-109F4 *Registration of Individuals and Review of Permitted Individuals* under NI 33-109;

PARSED DOCUMENT

	Document_Type	Document_Code	Document_Name	Part_Number	Part_Title	Content
3	Multilateral Instrument	11-102	Passport System	PART 4B	APPLICATION TO BECOME A DESIGNATED RATING	ORGANIZATION For the purposes of this Part, t...
4	Multilateral Instrument	11-102	Passport System	PART 4C	APPLICATION TO CEASE TO BE A REPORTING ISSUER	4C.1 Saskatchewan, Manitoba, Ontario, Qubec, ...
2	Multilateral Instrument	11-102	Passport System	PART 4A	REGISTRATION	4A.1 (1) (1) (1) (1) (1) (1) (1) (1) ...
1	Multilateral Instrument	11-102	Passport System	PART 4	DISCRETIONARY EXEMPTIONS	4.1 Specified jurisdiction For the purposes ...
0	Multilateral Instrument	11-102	Passport System	PART 3	PROSPECTUS	3.1 Principal regulator for prospectus 3.1 P...
5	Multilateral Instrument	11-102	Passport System	PART 5	EFFECTIVE DATE	5.1 Effective date 9, 2016, June 23, 2016, F...

Figure 14: Conversion from PDF to structured data using Template-1 script

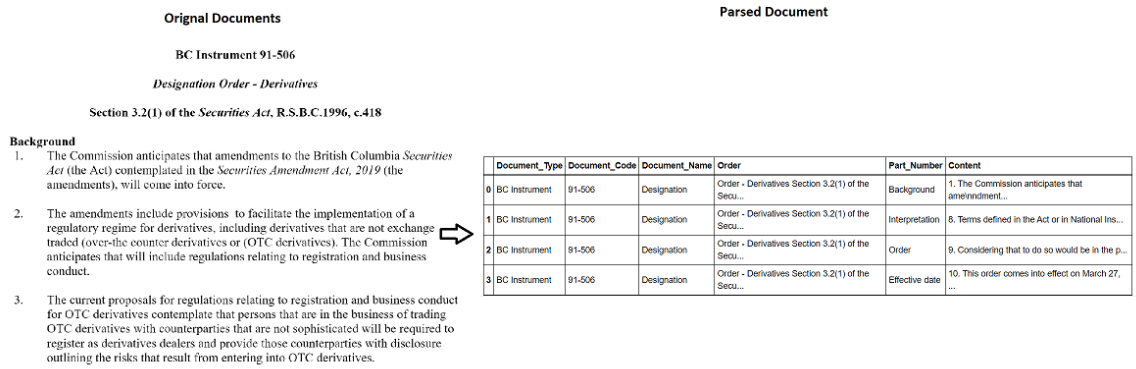


Figure 15: Conversion from PDF to structured data using template-2 script

Data Cleaning

The contents of all columns were cleaned and pre-processed with a developed python script "Data_Preprocessing_ analyze BC_Law_Dataset.ipynb", a visual explanation of the data cleaning algorithm can be seen in Figure 16. At first, we load the dataset to see if the columns and rows match with the British Columbia Securities dataset. Next, we values stored inside each column. After exploring the dataset, we standardize the column values. The column names such as Document Name, Document Type, Document Code had non-standard names. For example, "BC Instrument" and "BC INSTRUMENT" were all standardized to contain same standard names. Some of the document codes with Companion policy had keyword "CP" missing from their document, which was later corrected in the columns. All the irrelevant non- ASCII characters and the page numbers were removed from the were removed from the content column. For more details refer to the script called "Data_Preprocessing_BC_Law_Dataset.ipynb" inside the Lexata's GitHub Repository.

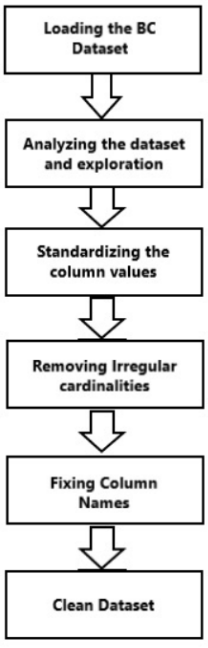


Figure 16: Algorithm for data cleaning

British Columbia Data set Description

The British Columbia Securities Law dataset is called “bc_law_processed.csv”. The dataset is formed after parsing Sections 1,3,6,7,8 and 9. It contains 2194 and 8 Columns. Figure 17 shows the head of the cleaned dataset.

The 8 Columns are:

- “document_type” contains the document type which can be National Instrument,
- BC Instrument, Companion Policy, Multi-lateral Instrument and other values
- “document_name” contains the document name, e.g. Passport System
- “document_code” contains the document code. In BC Securities Law denotes each type of
- Document with a unique code. Codes with a “CP” in the end indicate Companion Policy to the original document
- “date” indicates the date of the document when the document was published or when it is effective from. Document 81-106 has 2 documents. One document has the following date “Unofficial consideration for financial years beginning before January 1,2014”. Another document has the following date “Unofficial consideration for financial years after January 1,2014”
- “part” is a meta-tag that contains the part number and description. If the “part” values are empty “<HEADER_INFO>” is used to indicate header information.
- “section” is a meta-tag that contains the section number and description
- “content” contains the actual content of the document.
- “Flag” is a column used to indicate if a column contains table or images

	document_type	document_name	document_code	date	part	section	content	Flags
0	MULTILATERAL INSTRUMENT	Passport System	11-102	nan	PART 1 DEFINITIONS	1.1 Definitions	In this Instrument,\ncategory means a category...	NaN
1	MULTILATERAL INSTRUMENT	Passport System	11-102	nan	PART 1 DEFINITIONS	1.2 Language of documents - Québec	In Qubec, nothing in this Instrument shall be...	NaN
2	MULTILATERAL INSTRUMENT	Passport System	11-102	nan	PART 3 PROSPECTUS	3.1 Principal regulator for prospectus	(1) For the purposes of this section, the spec...	NaN
3	MULTILATERAL INSTRUMENT	Passport System	11-102	nan	PART 4 DISCRETIONARY EXEMPTIONS	3.2 Discretionary change of principal regulato...	If a person or company receives written notic...	NaN
4	MULTILATERAL INSTRUMENT	Passport System	11-102	nan	PART 4 DISCRETIONARY EXEMPTIONS	3.3 Deemed issuance of receipt	(1) Subject to section 3.5(1), a receipt for a...	NaN
5	MULTILATERAL INSTRUMENT	Passport System	11-102	nan	PART 4 DISCRETIONARY EXEMPTIONS	3.5 Transition	for section 3.3 (1) Section 3.3(1) does not ap...	NaN
6	MULTILATERAL INSTRUMENT	Passport System	11-102	nan	PART 4 DISCRETIONARY EXEMPTIONS	4.1 Specified jurisdiction	For the purposes of this Part, the specified j...	NaN
7	MULTILATERAL INSTRUMENT	Passport System	11-102	nan	PART 4 DISCRETIONARY EXEMPTIONS	4.2 Principal regulator	general Subject to sections 4.3 to 4.6, the pr...	NaN
8	MULTILATERAL INSTRUMENT	Passport System	11-102	nan	PART 4 DISCRETIONARY EXEMPTIONS	4.3 Principal regulator	exemptions related to insider reporting and ta...	NaN
9	MULTILATERAL INSTRUMENT	Passport System	11-102	nan	PART 4 DISCRETIONARY EXEMPTIONS	4.4 Principal regulator –	head office not in a specified jurisdiction Su...	NaN

Figure 17: Sample of British Columbia dataset

Ontario Specific Security Law

The data is collected from Ontario Securities Commission website [3] which is in pdf format. Navigate to Security laws -> Instrument, Rules and Policies to find all the documents based on the following categories:

- Categories
    - Procedure and Related Matters
    - Certain Capital Market Participants
    - Registration Requirements and Related Matters
    - Distribution Requirements
    - Ongoing Requirements for Issuers and Insiders
    - Take-Over Bids and Special Transactions
    - Securities Transactions Outside the Jurisdiction
    - Investment Funds
    - Derivatives

The important list of pdfs for the current product was provided by Leslie and consists of 137 document URLs (Figure-18).

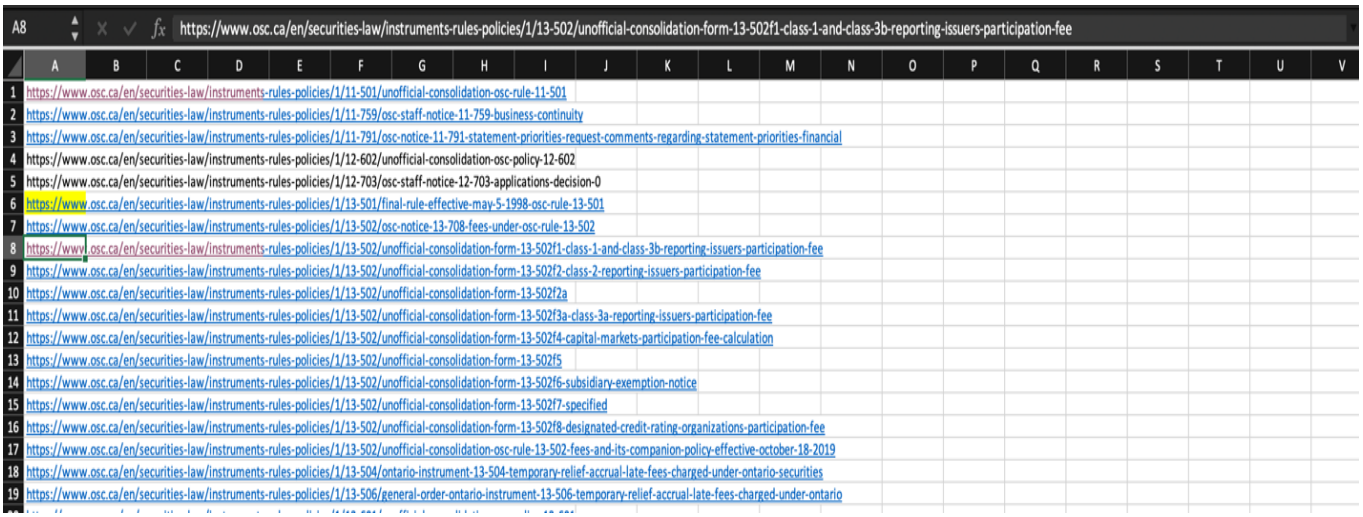


Figure 18: List of URLs

The list contains various format of pdf and can be divided into two formats i.e.,

a. Type 1:

The Type 1 do not consist of any forms or blank lines to fill and starts with contents in the document. It can have list of parts and sections or table of content (Figure 19).

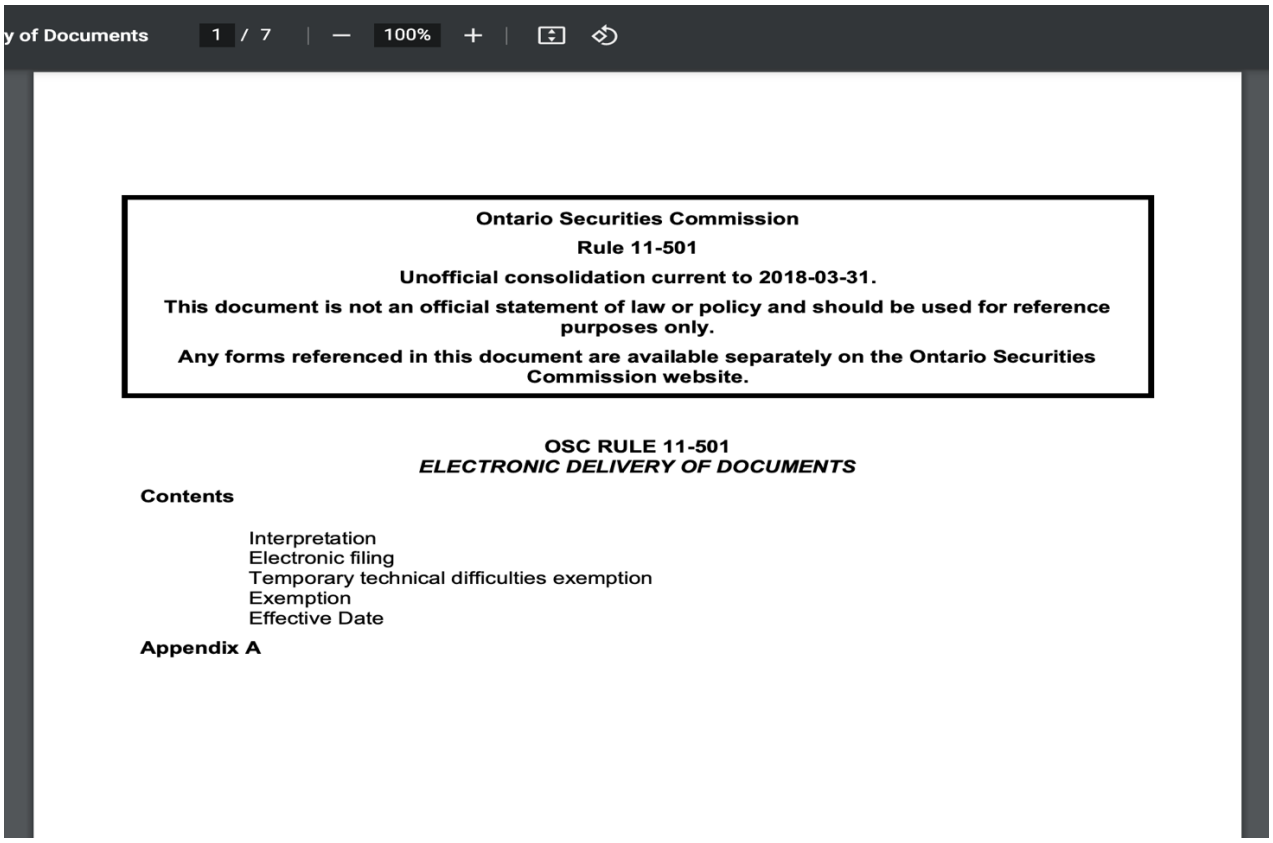


Figure 19: Sample of type-1 document

b. Type 2:

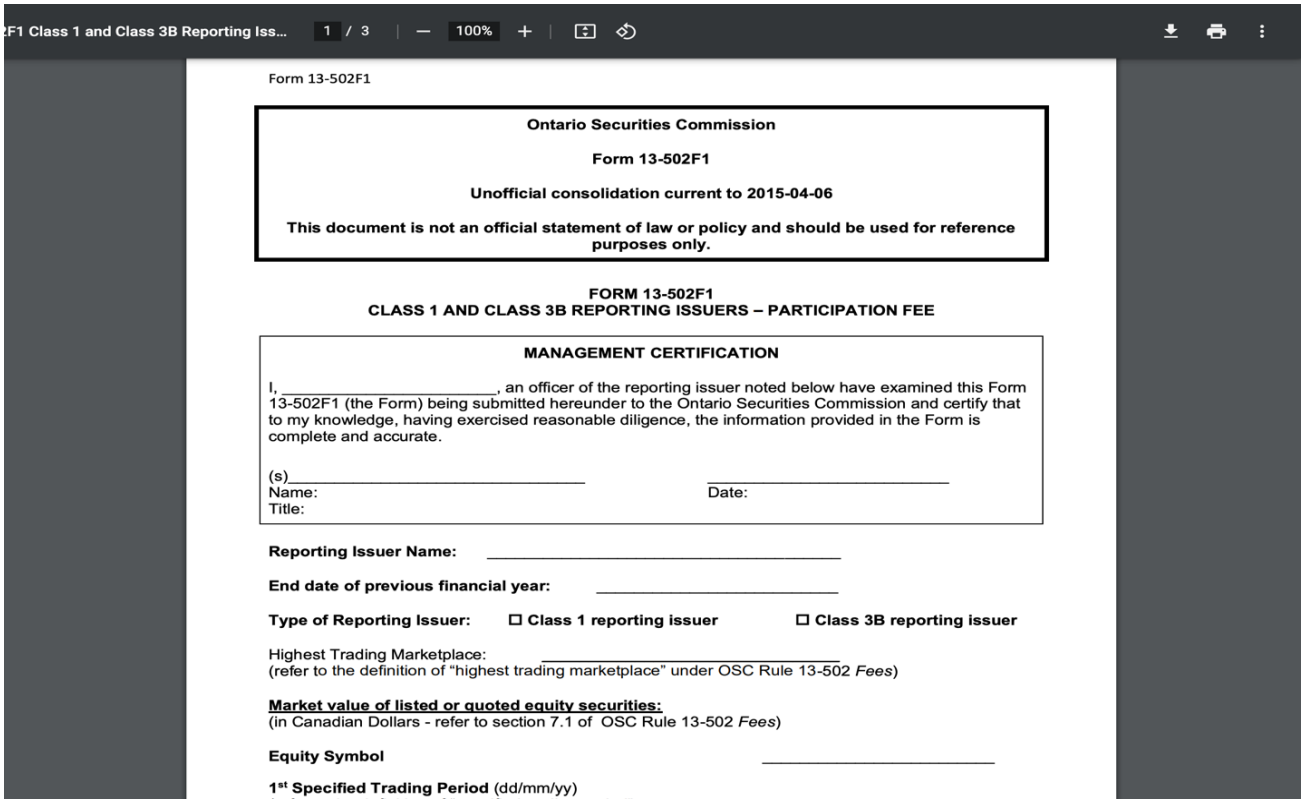


Figure 20: Sample of type-2 document

The Type-2 pdfs consists of forms or the blank spaces which require manual modification after the extraction and cleaning process is done (Figure 20).

The data from all the pdf documents should be extracted and parsed based on the Document Name, Document Type, Document code, File name, Section and its Contents. The final goal is to get a csv with all the above description as columns and clean it in order to use it in GPT- 3 open AI. We will be extracting the data into csv and convert to json format to our API in the project.

The pdf can be downloaded into drive accordingly to a drive and the python script [5] i.e. pdf_to_csv.ipynb can be applied on each pdf. Fitz library in PyMuPDF package reads pdf line by line and with the help of pandas and regex the headers/section and contents are separated. The separated columns are loaded to csv with respective category names. Example: csv_dataset_1.csv or csv_dataset_5.csv.

The saved csv is read into Ontariospecific_data_cleaning.ipynb python script which adds the Document Name, Document Type, Document code, File name and removes the header tags and content tags in the csv and also combines the same header names into one deleting all the duplicates. Finally, the cleaned csv (Figure 21) can be obtained.

Sheet 1

ontariospecific_csv_61-101CP

	Document Name	document_type	document_code	document_source	file_name	section	Content
2	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	Ontario Securities Commission Companion Policy 61-101CP	
3	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	Unofficial consolidation current to 2016-05-09. This document is not an official statement of law or policy and should be used for reference purposes only. Any forms referenced in this document are available separately on the Ontario Securities Commission website.	
4	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	COMPANION POLICY 61-101CP PROTECTION OF MINORITY SECURITY HOLDERS IN SPECIAL TRANSACTIONS	
5	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	Contents	
6	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	Part 1 General	General
7	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	Part 2 Interpretation	Equal Treatment of Security Holders Equity Participation by a Related Party Direct or Indirect Parties to a Transaction Amalgamations Transactions Involving More than One Reporting Issuer Previous Arm's Length Negotiations Exemption Connected Transactions Time of Agreement "Acquire the Issuer"
8	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	Part 3 Minority Approval	Meeting Requirement Second Step Business Combination Following an Unsolicited Take-over B Special Circumstances
9	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	Part 4 Disclosure	Insider Bids – Disclosure Business Combinations and Related Party Transactions – Disclosure
10	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	Part 5 Formal Valuations	General Independent Valuators
11	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	Part 6 Role of Directors	Role of Directors
12	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	PART 1	
13	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	GENERAL	
14	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	General	
15	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	1.1	The Autorité des marchés financiers and the Ontario Securities Commis regard it as essential, in connection with the disclosure, valuation, review processes followed for insider bids, issuer bids, business combinations ar transactions, that all security holders be treated in a manner that is fair an be fair. We are of the view that issuers and others who benefit from access markets assume an obligation to treat security holders fairly, and that the obligation is essential to the protection of the public interest in maintaining operate efficiently, fairly and with integrity. We do not consider that the types of transactions covered by this Instrun unfair. We recognize, however, that these transactions are capable of bein and have made the Instrument to address this. This Policy expresses our views on certain matters related to the Instrum
16	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	PART 2	
17	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	INTERPRETATION	
18	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	Equal Treatment of Security Holders	
19	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	2.1 (1)Security Holder Choice	– The definitions of business combination, collateral benefit and interested party, as well as other provisions in the Instrument, include identical treatment of security holders in a transaction. For the purposes o security holders have an identical opportunity under a transaction, then th be treated identically. For example, if under the terms of a business combi holder has the choice of receiving, for each affected security, either $10 in share of ABC Co., we regard the security holders as having identical entiti form, and as receiving identical treatment, even though they may not all m This interpretation also applies where the Instrument refers to considerati equal in value" and "in the same form", such as in the provisions on secur combinations.
	Security Law	Instruments,Rules and policies	6	OSC	Ontario Securities Commission Companion Policy 61-101CP	(2)	– The definitions of business combination and

Figure 21: Cleaned csv dataset

Automated Parsing

A package known as PyMuPDF is downloaded to access its library such as Fitz. After importing necessary libraries few user defined functions or definitions are coded i.e., def fonts which identifies font size and font style of each word in each line of the pdf (Figure 22).

```
In [2]: from google.colab import drive
        drive.mount('drive')
        DrivPath='drive/My Drive/Lexata/Dataset2/1/'

        Mounted at drive

In [3]: pip install PyMuPDF==1.16.14

        Collecting PyMuPDF==1.16.14
          Downloading https://files.pythonhosted.org/packages/64/6d/e8f9cd7748ada73f34b9e92f4b3cd840be332f5c50bc7edc1ebd0c6dba98/PyMuP
        DF-1.16.14-cp37-cp37m-manylinux2010_x86_64.whl (5.7MB)
             |████████████████████████████████| 5.7MB 3.9MB/s
        Installing collected packages: PyMuPDF
        Successfully installed PyMuPDF-1.16.14

In [4]: import csv
        from operator import itemgetter
        import fitz
        import json
```

Figure 22: Explanation of PyMuPDF

Later, each pdf is read and <h0> and <p/content> tags are given based on the style and size of the words. With the help of pandas and regex the tags are separated into two different columns. The Figure-23 is an example of the csv file created.

```
df.head(30)
```

Out[14]:

	0	Content
0	<h0>Decisions, Orders and Rulings \n	0
1	<h0>Decisions, Orders and Rulings \n	\n
2	<h0>Decisions, Orders and Rulings \n	\n
3	<h0>Decisions, Orders and Rulings \n	\nApril 16, 2020 \n\n
4	<h0>Decisions, Orders and Rulings \n	\n(2020), 43 OSCB 3661 \n
5	<h0>Decisions, Orders and Rulings \n	\n
6	<h0>2.2.3 \nOntario Instrument 31-512 Relief i...	0
7	<h0>Relief in respect of Client Focused Reform...	0
8	<h0>Relief in respect of Client Focused Reform...	<p/content>The Ontario Securities Commission, ...
9	<h0>Relief in respect of Client Focused Reform...	<p/content>Registration Requirements, Exemptio...
10	<h0>Relief in respect of Client Focused Reform...	<p/content>" is made, to provide relief to reg...
11	<h0>Relief in respect of Client Focused Reform...	<p/content>Registration Requirements, Exemptio...
12	<h0>Relief in respect of Client Focused Reform...	<p/content>. \n\nApril 15, 2020 \n\n"Grant Vin...
13	<h0>Authority under which the order is made: ...	0
14	<h0>Authority under which the order is made: ...	<p/content>Act and section:
15	<h0>Authority under which the order is made: ...	<p/content>Securities Act
16	<h0>Authority under which the order is made: ...	<p/content>, subsection 143.11(2) \n\n
17	<h0>Decisions, Orders and Rulings \n	0
18	<h0>Decisions, Orders and Rulings \n	\n
19	<h0>Decisions, Orders and Rulings \n	\n
20	<h0>Decisions, Orders and Rulings \n	\nApril 16, 2020 \n\n
21	<h0>Decisions, Orders and Rulings \n	\n(2020), 43 OSCB 3662 \n
22	<h0>Decisions, Orders and Rulings \n	\n
23	<h0>Ontario Securities Commission \n\nOntario ...	0
24	<h0>Relief in respect of Client Focused Reform...	0
25	<h0>Definitions \n\n	0

Figure 23: The csv data with headers and content (csv_dataset_6.csv)

Data Cleaning

The csv document created from above pdf_to_csv.ipynb[5] python script is read into Ontariospecific_data_cleaning.ipynb script and the columns are renamed as section and content. The repeated section rows are grouped and concatenated along with the removal of any duplicate headers. We will be using pandas and numpy libraries to perform all the above activities (Figure24).

```
Mounted at drive

In [46]:
#import the necessary packages and read csv
import io
import pandas as pd
import numpy as np
df = pd.read_csv(DrivPath+'csv_dataset_6.csv')

In [47]:
# Rename the columns
df1 = df.rename(columns = {'Unnamed: 0': 'Slno', '0': 'section'}, inplace = False)

In [49]:
# Cross check the headers
df1
```

Out[49]:

	Slno	section	Content
0	0	<h0>4. \nFailure to comply with any other requ...	\n
1	1	<h0>July 27, 2017 \n	0
2	2	<h0>July 27, 2017 \n	\n\n
3	3	<h0>July 27, 2017 \n	\n(2017), 40 OSCB 6577 \n
4	4	<h0>July 27, 2017 \n	\n
...	...	...	...
253	253	<h0>Notices / News Releases \n	<p/content>Lanion Beck \nSenior Legal Counsel ...
254	254	<h0>Notices / News Releases \n	<p/content>The Manitoba Securities Commission ...
255	255	<h0>Notices / News Releases \n	<p/content>Chris Besko \nDirector, General Cou...
256	256	<h0>Notices / News Releases \n	<p/content>Financial and Consumer Services Com...
257	257	<h0>Notices / News Releases \n	<p/content>Jason Alcorn \nSenior Legal Counse...

258 rows × 3 columns

```
In [50]:
# Check for column names
df1.columns

Out[50]: Index(['Slno', 'section', 'Content'], dtype='object')

In [51]:
# concatenate the string
df1['Content'] = df1.groupby(['section'])['Content'].transform(lambda x : ' '.join(x))

In [52]:
# drop duplicate data
```

Figure 24: csv to Pandas Data frame

Later the Document Name, Document Type, Document code, File name to respective headers and contents area added to the data frame. The <h0>, <p/content>, 0, \n and NULL values are removed. The Document Name, Document Type, Document code, File name can be modified or changed accordingly to each document code or File name. The unique values are checked to compare the data frames (Figure 24).

Example:

Document Name: Security Law, Document Type: Instruments, Rules and policies, Document code: 6 (Can be 1,3,4,5,6,7,8 or 9), File name: Ontario Securities Commission Companion Policy 61-101CP (It can be any other document name), Section: Contains Headers including part, section and headings, Content: Contains content or paragraph of the header part or section.

```
In [54]:
# Replace \n in the dataframe
df_c1=df1.replace({'\n': ' '}, regex=True)

In [55]:
# Replace <h0> tag in the dataframe
df_c2=df_c1.replace({'<h0>': ' '}, regex=True)

In [56]:
# Replace <p/content> tag in the dataframe
df_c3=df_c2.replace({'<p/content>': ' '}, regex=True)

In [57]:
# Replace 0 in the dataframe
df_cleaned=df_c3.replace({'0': ' '}, regex=True)

In [58]:
# Inserting new columns into the dataframe
# Can change Document_code and File_Name
df_cleaned.insert(1,"Document_Name","Security Law")
df_cleaned.insert(2,"Document_Type","Instruments,Rules and policies")
df_cleaned.insert(3,"Document_code","6")
df_cleaned.insert(4,"Document_source","OSC")
df_cleaned.insert(5,"File_Name","Notices / News Releases:Multilateral CSA Staff Notice 61-302 Staff Review and Commentary on M
ultilateral Instrument 61-101 Protection of Minority Security Holders in Special Transactions")

In [59]:
# Check the dataframe
df_cleaned
```

Out[59]:

	Slno	Document_Name	Document_Type	Document_code	Document_source	File_Name	section	Content
0	0	Security Law	Instruments,Rules and policies	6	OSC	Notices / News Releases:Multilateral CSA Staff...	4. Failure to comply with any other requirem...	
1	1	Security Law	Instruments,Rules and policies	6	OSC	Notices / News Releases:Multilateral CSA Staff...	July 27, 2 17	(2 17), 4 OSCB 6577
5	5	Security Law	Instruments,Rules and policies	6	OSC	Notices / News Releases:Multilateral CSA Staff...	Chapter 1	
6	6	Security Law	Instruments,Rules and policies	6	OSC	Notices / News Releases:Multilateral CSA Staff...	Notices / News Releases	
8	8	Security Law	Instruments,Rules and policies	6	OSC	Notices / News Releases:Multilateral CSA Staff...	1.1 Notices 1.1.1 Multilateral CSA Staff ...	

Figure 25: Final Cleaned Data Frame

The final cleaned data frame is written into csv and appended as the files are cleaned (Figure 25). The graphical representation of all the stages in data collection and pre-processing can be observed in the data pipeline description.

Data Modelling

Data Preparation Tasks:

Initial data sections and irregular cardinality checks

Before modelling the data for the semantic search functionality, we need to make sure to handle the irregular cardinalities and group the data that will be easier for the Open AI to structure and render on the Python framework.

At this point we have done the basic data preparations, however, to compare the functionality of the Open AI we needed data that is commonly present on our framework as well as on the existing Lexata's framework. Since, we have been provided with the parsed data with multiple sections (National Instruments, Multilateral instrument, Companion Policy etc), we cut the document and extract the 81-102, 81-106, 81-107 document codes for National instrument and upload the document for multi-engine search to perform critical analysis.

To automate this process, a python script runs in the backend which takes the complete parsed data as input and cuts the data on the mentioned sections and pushes the data on for next level parsing. Here, we observe that there is a possibility to handle the Unicode characters which might create an unnecessary confusion at the search optimization of the Open AI engine. Hence, we encoded and decoded these characters for a smooth and flawless rendering of the document.

This is an important feature in the automation process since we now know the flow and the contents of the document it can be an easier task to critically analyse the model and the search outcomes.

The data preparation steps extend its functionality after the single document base model, as we discover the limits of GPT-3 engines and then decide to cut the data into sections and produce the batches of the data and push these batches on GPT-3. A python script which runs in background takes the whole document as input containing more than 200 sections and returns the chunks or batches of data with data-frames having less than 200 instances. These data frames are then further used as an input for the search query of the Open AI.

Conversion in Data Dictionaries

As mentioned in the Open AI documentation file-based searching can be done using a key value pair. We follow 2 steps in this process to reduce the documents in the provided file with max_rerank value of 5. To make this step easy we use pandas to convert the existing csv inputs to Json key value pairs where in every document is indexed and the metadata is the contents of the document divided by sections. Figure 26 shows the sample of Json format of the document which contains the text and the contents sections. It can be observed that the text contains the parts that is the law sections of the documents and the metadata generated is the content of the document.

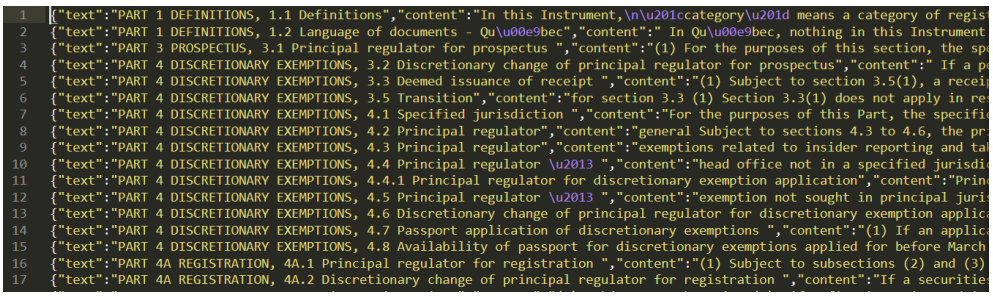


Figure 26: Sample Json format document.

An interpretable code snippet through python jupyter notebooks can be observed in the Figure 27.

	text	metadata
0	Document_1 acceptable summary form	acceptable summary form acceptable summary for...
1	Document_2 Initial report	Initial report A reporting insider must file a...

Figure 27: Document Metadata structure from python document

Python Framework

Since we have prepared the data, the next step will be to integrate it with the Open AI platform and create a virtual playground to analyse results. To create a framework, we use python Django environment to display and render the information. We have chosen Django environment as it is highly dynamic, and it contains model-template-views type architecture. This is a temporary platform that can be used to perform comparative analysis with the existing platform. Since, this platform is highly dynamic and the templates can be modified ahead in time as per the requirement of the organization, it provides scalability.

A simple python framework was created which displays the sections of the cut data which contains information as 81-102, 81-106, 81-107 document codes for National instrument. The framework renders three main pages, home page, about page and Login page. Apart from this, the framework also renders the search output page which is indeed the matched output of the search query depending on the selected engine and score value.

Login Page: This is the main landing page wherein we have included a secure login information so that only authorized users can be provided access to the existing search framework, as GPT-3 searching on open AI charges the owner for every search query. Refer Figure 28 to have a look at the login page.

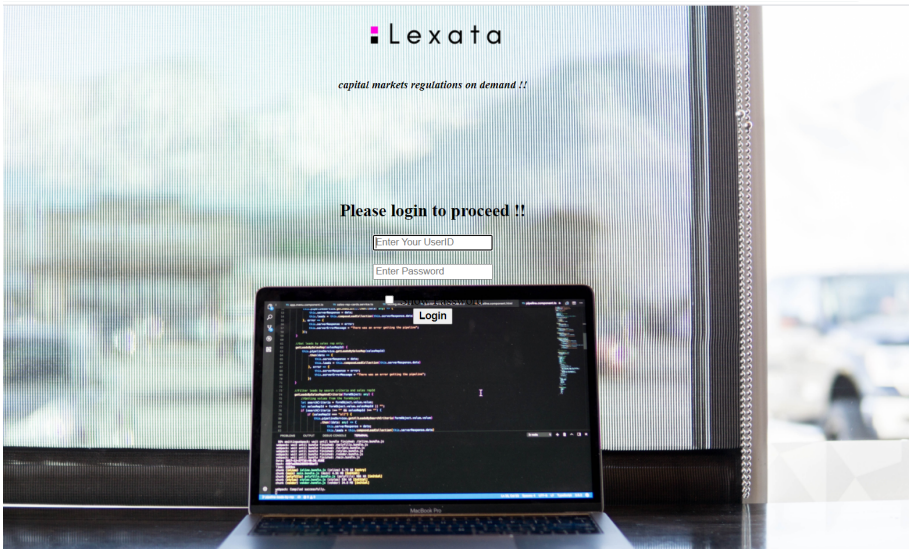


Figure 28: Snapshot of the Login Page

Home Page: A snapshot of home page can be observed in the below Figure 29. This page renders the document containing the sections as serves as the main page to traverse through the platform. It gives the options to choose the engines which are “Davinci”, “Ada”, “curie” and “Babbage”. It also displays the main contents of the sections on the right panel. This page gives a drop down to choose number of scores per page. It gives a redirection to the about page and the logout link.

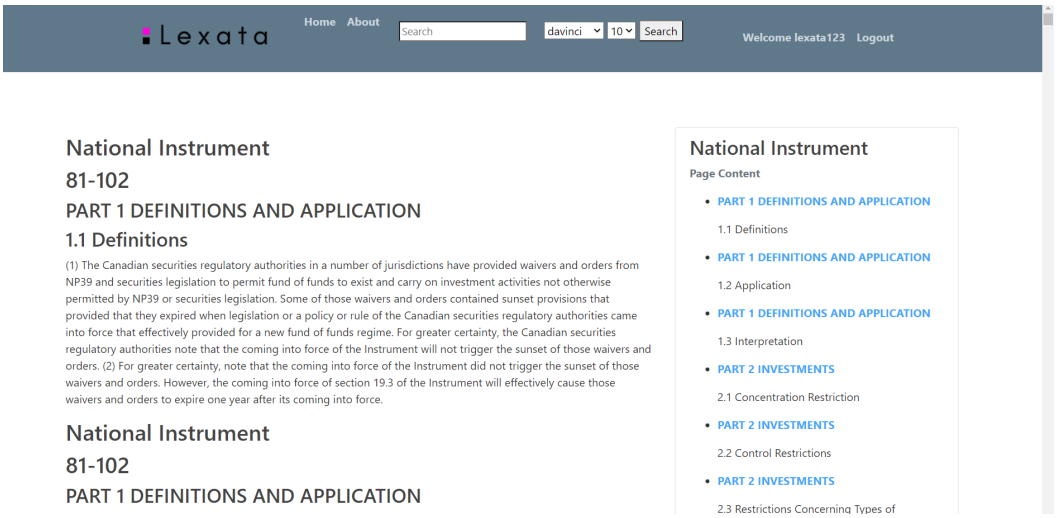


Figure 29: Snapshot of Home Page

About Page: This page displays general information about the owners and the purpose of this framework. A snapshot of the about page can be seen in the Figure 30.

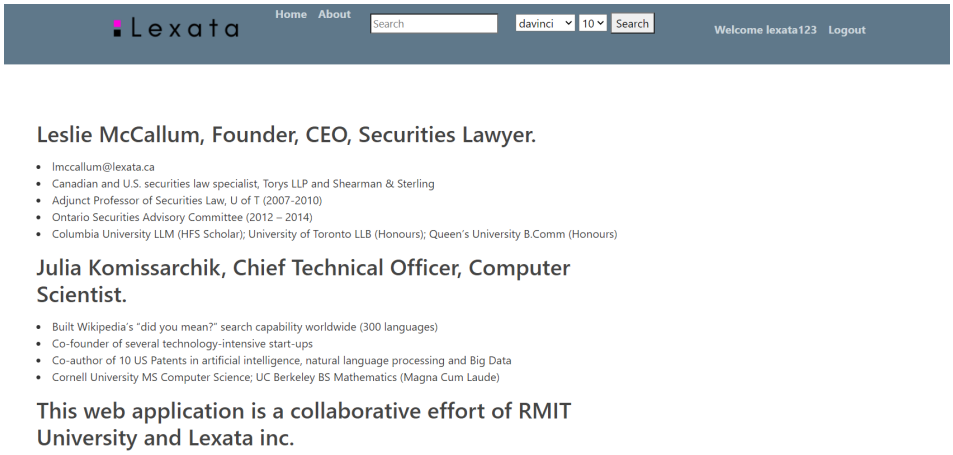


Figure 30: Snapshot of About Page

Integration of Open AI and GPT-3

In this section we will talk about the integration of GPT-3 with the created Django framework. This section covers a brief introduction of GPT-3 the working and the outputs.

Open AI API: Open AI API provides a general-purpose "text in"- "text out" interfaces, this is different from the other language models which are designed to perform just a single task.

Introduction to GPT-3

GPT-3 is a generative pre-trained Transformer that has been pretrained on 175 billion parameters was introduced by Open AI and published under Cornell University on 28th May 2020. GPT-3 is an extension of the previous generative training models, and it is the third-generation language prediction model which used deep learning techniques to predict, search and classify information that is more associated with the human centric decision [6].

GPT-3 is an autoregressive model which has been trained on a large corpus of data and can be applied on the required tasks without any gradient updates or fine tunings. The few-shot learning demonstrations are specified purely via text interactions with the model. This model has been trained to generalize well with all the information provided and learn and detect anomalies and insights in the data which can be used for various purposes [7].

Significance of GPT-3 in Lexata

The approach of transferring information from the pretrained models by freezing the weights and combining the applied transformations on the task specific corpus of data is exhaustive. This pertains to the fact that the pre-trained model might need the fine tuning on the task specific publications to reduce loss and increase throughput. However, as specified in the data collection and preparation steps the client had a requirement of training a model on the huge corpus of legal data that can be used to perform semantic searching amongst various sections of data.

In addition, for the above-mentioned challenge, it was crucial for us to understand the correlation between the provided information for the sematic and comparison analysis. Hence, the model should exploit the spurious correlation between the provided data corpus and generalize well with the outcomes.

Semantic searching technique anticipates the requirement of broad sets of capabilities to recognize patterns and generalize inputs to rapidly adapt and recognize the given task. The language model of GPT-3 has an ability of "in-context learning". Which indeed means that at every forward pass in the neural network the model learns and adapts to the sequences and patterns to embed the repeated subtasks during the training [7].

A visual representation of these patterns can be seen in the below Figure 31. The synonyms detection of patterns can be seen for some legal terms in the provided data corpus.

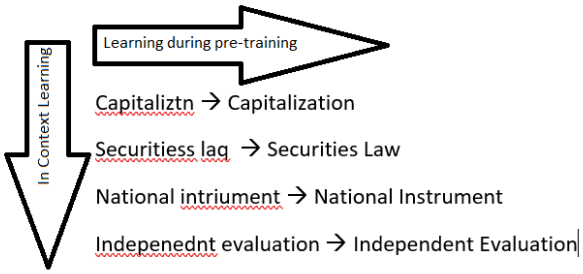


Figure 31: In-context and pre-training learning

Soon after the release of GPT-3 there has been numerous blogs and publications where possibilities of GPT-3 on various legal data summarization and abstraction tasks were discussed [8]. Since our client had the access of Open AI API for integration of GPT-3 and to adhere these discussions, we continued to check the adaptivity of this model for the performance on our corpus of data.

Base Model using GPT-3

Base model preparation was performed by limiting the training information and observing the single search characteristics. Abilities of GPT-3 to perform this task was explored in the initial development of this project. Semantic searching methodology differentiates from the lexical searches by training the algorithm to cluster data based on the relevance and produce the most relevant outcomes instead of matching the literal meaning of the query words [9]. Therefore, to implement this requirement we used GPT-3 to understand the Natural Language, process, and cluster the same and provide results that match with the relevance.

To start off with the integration of GPT-3, a python script was written to pass the converted Json file object to Open AI with a purpose set as "search". This file had the key value pair as discussed in the data preparation. Once the GPT-3 was trained on this file, a response function was triggered with parameters such as the engine type, search query, max_rerank, uploaded file ID. In the initial training phase, we chose to go with the "ada" engine and the max_rerank value as 5. The max_rerank includes all documents that are re-ranked by the semantic search, the default value is 200, however, the higher value leads to increased cost. Hence, we set a recommended value as described in open AI documentation [6]. The file ID is a unique identification value generated when the document is uploaded for training.

Upon analysing the response for the query, it was found the most relevant section was returned as an output by the GPT-3. The returned section was indeed the key for the value in the content of the passed document. The relevancy score was returned with the output as well.

The Figure 32 below shows a snippet of the GPT-3 output which returns data as well as model information with relevancy score.

```
response
```

```
<OpenAIObject list at 0x7fdfc0da3270> JSON: {
  "data": [
    {
      "document": 0,
      "object": "search_result",
      "score": 6.553,
      "text": "Document_2 Initial report"
    }
  ],
  "model": "ada:2020-05-03",
  "object": "list"
}
```

Figure 32: Single Document Search Response

This response was further tested to observe the results for different engines and other parameters; however, these functionalities were explored under the client's guidance and response as we were aware that all search results with different parameters were indeed costing the client. After analysing our base model, we concluded that the response from GPT-3 was somewhat relevant, and the functionalities can be explored further by training GPT-3 on multiple documents. Once after training the model on multiple documents the search query can be used to return multiple relevancy scores that can be useful for us to achieve our aim of semantic searching.

Incremental Model using GPT-3

Since we had developed our base model using the Open AI GPT-3 model, in this section we aim to extend and explore our capabilities using multiple documents and factoring the scores accordingly. As we aim to produce semantic search results, we need to cluster the documents based on their relevancy and display the most relevant sections with its content. A visual representation of this task can be observed in Figure 33 [10].

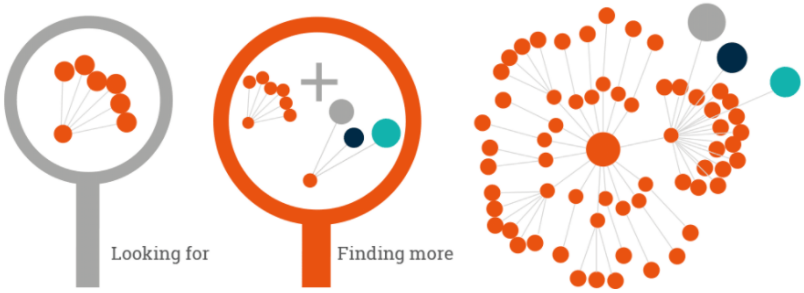


Figure 33: A Visual Representation of Semantic Search

To cluster the scores, we trained GPT-3 with multiple documents. Since we needed the sections to be the key while searching the contents in the Json file, we explored and found that some contents have replicated values for sections and there existing abnormalities in the pattern of sections. Moreover, we also found that

open AI restricts the training of GPT-3 to a maximum of 200 documents. Hence, to overcome these challenges, we came up with a strategy to combine the "part" and "section" to form a composite key having the value as the content. Since "part" is an upper-level segregation of the document and "sections" are its subdivision, hence, both together serves our purpose of representing the key. To overcome the training document restrictions of Open AI, we wrote a python script which takes the whole document as the input and returns the chunks or batches of data with data-frames having less than 200 instances. These data frames have been further used to train GPT-3.

This was a useful output as the response of the results can now be captured on multiple documents and can be used for comparison. The response data frame contains the score values which displays the relevance of the search query as we are performing the semantic searching using the Open AI.

This information is then rendered on Django framework, although we are not displaying the score on the website to maintain the aesthetics, the score values however can be checked on the exported csv files.

Thus, the sematic searching was now established using the GPT-3 model and the highest relevancy scores can be grouped together. To increase the accessibility of the search relevancy functions, we have provided the dropdown on the python framework as explained in the above sections. This allows the user to test the search query results based on the choice of top 10,20 and 30 relevancy scores. A snapshot of the response scores with the relevant content can be observed in Figure 34.

	document_name	document_type	document_code	document_source	part	section	content	combined	score
0	National Policy	Process for Registration in Multiple Jurisdict...	11-204	BC	PART 2 DEFINITIONS	2.3 Interpretation	Unless the context indicates otherwise, a refe...	PART 2 DEFINITIONS, 2.3 Interpretation	555.744
0	National Policy	Process for Prospectus Reviews in Multiple Jur...	11-202	BC	PART 2 DEFINITIONS	2.2 Further definitions –	Terms used in this policy and that are defined...	PART 2 DEFINITIONS, 2.2 Further definitions –	549.979
1	National Policy	Electronic Delivery of Documents	11-201	BC	PART 1 – GENERAL	1.1 Definitions –	In this Policy "delivered" means transmitted, ...	PART 1 – GENERAL, 1.1 Definitions –	549.234
2	National Policy	Process for Exemptive Relief Applications in M...	11-203	BC	PART 2 DEFINITIONS	2.2 Further definitions –	Terms used in this policy that are defined in ...	PART 2 DEFINITIONS, 2.2 Further definitions –	548.576
3	Companion Policy	Passport System	11-102CP	BC	PART 1 GENERAL	1.1 Definitions	In this Policy, "CP 33-109" means Companion Po...	PART 1 GENERAL, 1.1 Definitions	546.564
4	National Policy	Process for Prospectus Reviews in Multiple Jur...	11-202	BC	PART 1 APPLICATION	1.1 Scope and application –	This policy describes procedures for the filin...	PART 1 APPLICATION, 1.1 Scope and application –	544.545
5	Multilateral Instrument	Passport System	11-102	BC	PART 1 DEFINITIONS	1.1 Definitions	In this Instrument,\n"category" means a catego...	PART 1 DEFINITIONS, 1.1 Definitions	537.730

Figure 34: Relevancy score ranked document

Critical Analysis of the Model

The sematic search technique was integrated in the existing framework and the results were looking good, however, to analyse our existing model and comparing the same with different engines could benefit us to understand the difference between the returned scores and the most stable engine to deploy.

Numerous different engines provided by the open AI platform, to explore this feature, we included the engine selections and rendered the outputs based on the different engines and the outputs are captured in the respective CSVs with their respective responses. As mentioned earlier the responses include the individual query search results and their scores.

Open AI interface provides access to several different engines: Ada, Babbage, Curie and Davinci. The Open AI claims that each engine is configured to perform some specific tasks, they too claim that Davinci engine is best at overall performance. Although there is very little information to know more about the configurations of the engines and their respective parameters or pre-training information, it is claimed that the typical scores range in the values 0 to 300. It can go higher which is a sign of good sematic score.

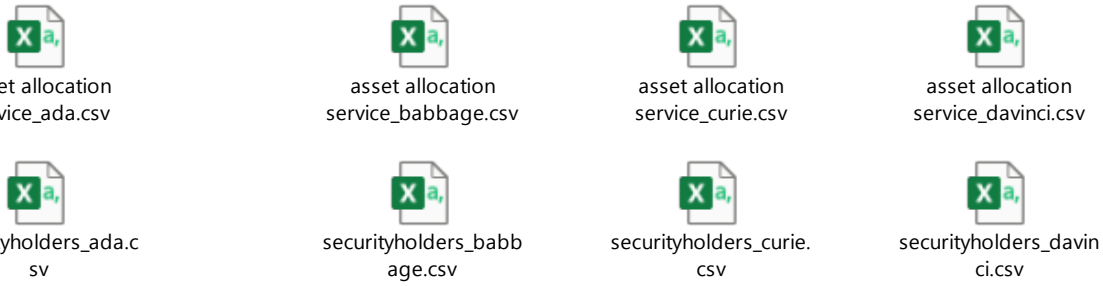
We based on our search query results we have typically observed that higher score values are obtained on the Davinci engine, and hence, we can claim that although Open AI claims that "Curie" is better when it comes to language translation, we have found that Davinci delivers better scores.

In the below sample output screenshot Figure 35, we can observe the document specific results that are returned by the open AI engine. It can be observed that we have the document number, the search result, and the respective score of the sematic search result.


```
<OpenAIObject list at 0x7fcfd06e7720> JSON: {
  "data": [
    {
      "document": 0,
      "object": "search_result",
      "score": 212.813
    },
    {
      "document": 1,
      "object": "search_result",
      "score": 54.52
    },
    {
      "document": 2,
      "object": "search_result",
      "score": 39.227
    }
  ],
  "model": "davinci:2020-05-03",
  "object": "list"
}
```


Figure 35: Document Specific results

The results of the response of each query were stored in files which were produced to a domain expert (Leslie McCallum). These files were interpreted, and the output results were judged on their relevance to the input query. This is done to make the comparison easier and structured. Here are some of the results CSV's that are captured. Below are some of the csv's that are returned as a response, these contain the information as well as the scores of the results.



These files are read as per the convention: ***'query_enginename'***

Model Evaluation

- As we have established that the response from the GPT-3 for every semantic search query contains a relevancy score matched with the query response. This score can be used to evaluate the model using different engines.
- As per the information provided by the Open AI about the engines the following can be deduced:
- Davinci → This engine has the highest computational capability and abilities to perform any task with minimum instructions. The costing of this engine is the highest, however it goes with the implacable computational capabilities. It is said to have the best summarization and in-context learning and content generation [11].
- Curie → It is not as strong as Davinci; however, it has good capabilities while analysing complicated text, sentiment analysis and summarization [11].
- Babbage → This engine is used or trained to perform more straightforward tasks. It has capabilities for semantic searching and costing is much lower than Davinci [11].
- Ada → Ada is a fast model that can be used to parsing and certain kinds of classification tasks [11].
- We concatenated all the top scores generated by the engines and plotted the difference in these scores to have the insights based on the query about the performance of these engines. In the below Figure 36, the plots can be observed.

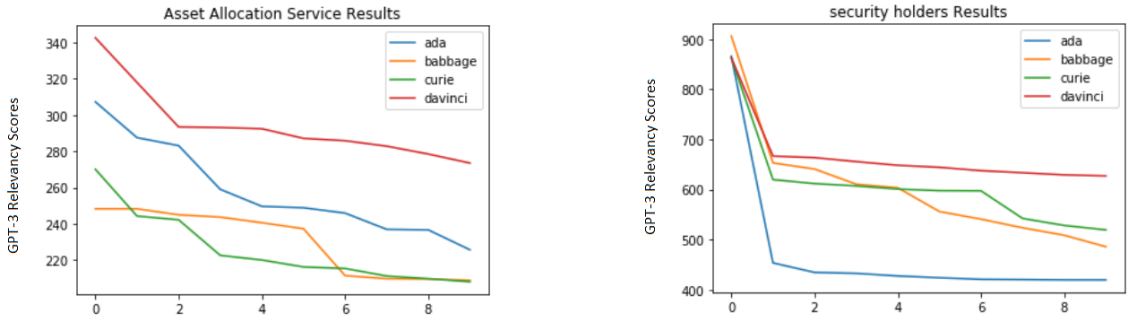


Figure 36: GPT-3 relevancy scores across multiple engines and query response

To evaluate the engines [12], multiple queries were fired, and results were observed and judged based on the relevancy scores. Two such query results for "Asset allocation service" and "Security Holders" were fired, and generated scores were captured and plotted in the Figure 36. It can be observed that Davinci engine performs the best on both the queries, but as per our expectations Babbage did not perform very well in few of the queries but performs exceptionally well on the second query. Other than Davinci and Babbage other engines performances were consistently low as expected. Hence, by analysing our model we can state that going forward Lexata can incorporate Babbage engine since it has much lower costing as compared to Davinci, and the performance of this engine is adequate as per the needs. A tabular information of the exact scores on both the queries is displayed in the Figure 37.

Asset Allocation Service

	ada	babbage	curie	davinci
0	307.262	248.305	270.075	342.588
1	287.550	248.289	244.291	317.978
2	283.147	244.997	242.234	293.445
3	259.055	243.722	222.637	293.150
4	249.657	240.629	220.073	292.470
5	248.890	237.324	216.247	287.157
6	245.929	211.458	215.402	285.855
7	236.981	209.759	211.226	282.851
8	236.673	209.544	209.778	278.508
9	225.706	208.879	208.080	273.532

Security Holders

	ada	babbage	curie	davinci
0	866.068	906.378	862.468	862.933
1	453.593	653.173	619.781	666.809
2	434.608	640.923	611.852	663.691
3	432.689	610.668	607.226	655.779
4	427.541	603.465	600.974	648.419
5	423.963	556.115	597.752	644.369
6	420.817	540.878	597.392	637.685
7	420.327	523.659	542.353	633.541
8	419.672	508.782	528.374	629.238
9	419.557	485.936	519.457	627.210

Figure 37: Top 10 scores comparison across all engines

Comparative Analysis

As mentioned in the framework design, we aim to make a platform that is identical to the existing framework as well as is scalable and robust. This includes integration of GPT-3[13] on our python framework and rendering the outputs as the response of the queries. The below Figure 38 describes the comparison of existing and the proposed platform.

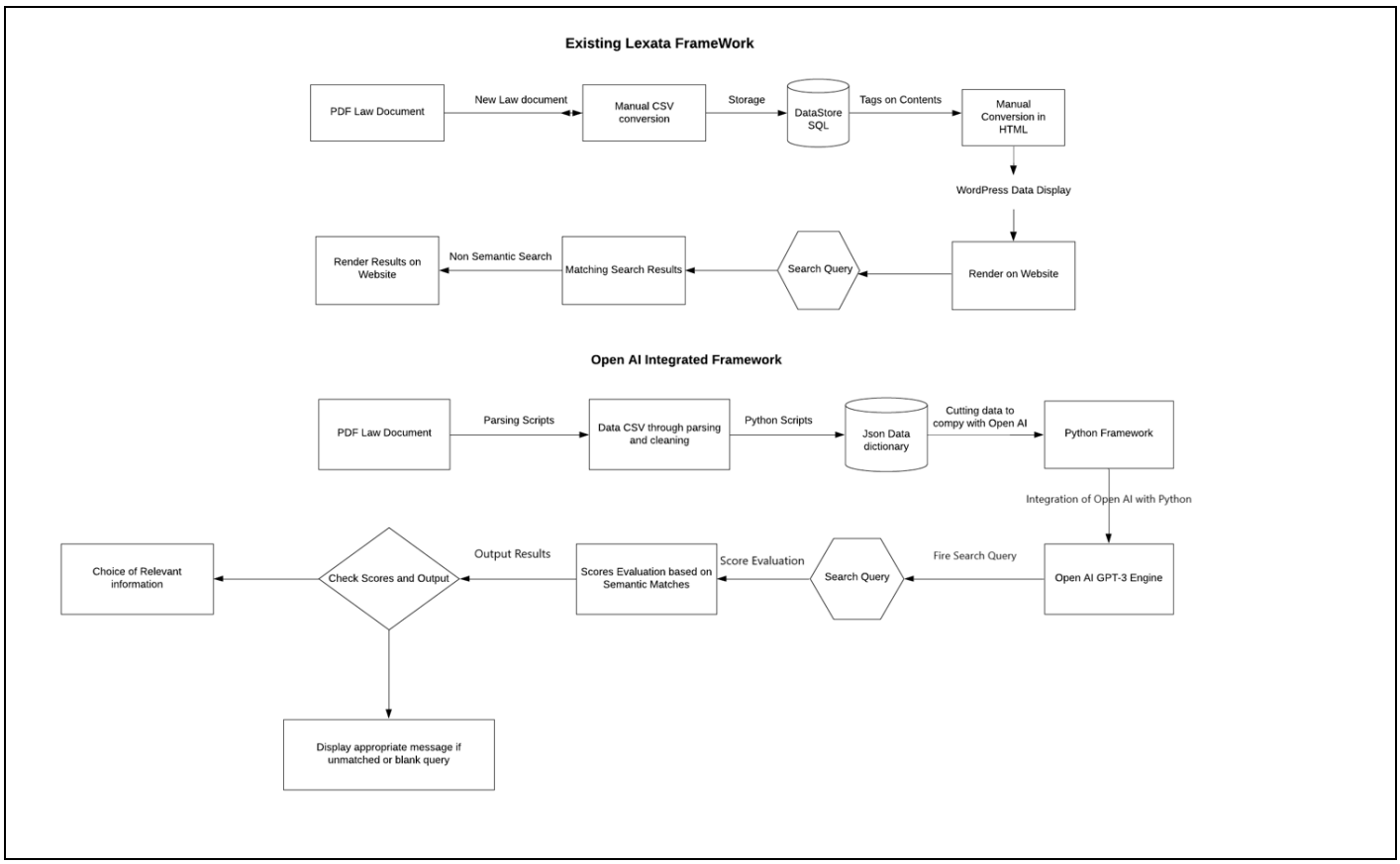


Figure 38: Comparative Analysis of existing and new model

The structural differences can be observed in the above figure. The existing framework rendered information based on the lexical search methods which involved use of basic searching algorithms through the database and produce the results based on keyword match.

The proposed platform is an end-to-end product which divides the data into chunks and performs semantic searching based on scores and relevancy to display the most appropriate information and renders the details on the web platform. This structure was proposed based on the initial requirement of the client while performing iterative development with the base model. Figure 39 displays the initial client requirement.

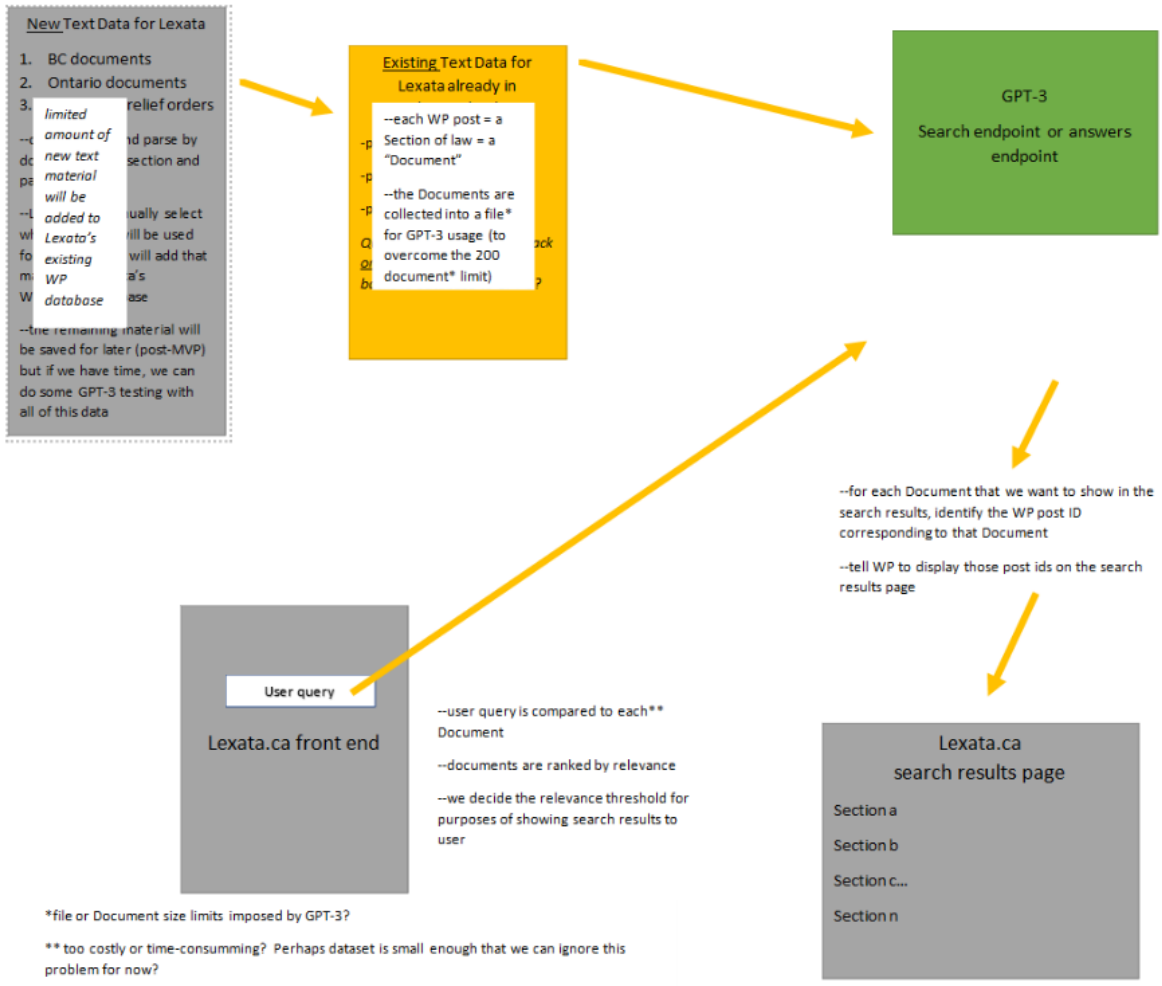


Figure 39: Initial Client requirement

App deployment of MVP on AWS cloud platform

As we can see in the Figure 40 given above **application.py** file is the main file containing the python code for website. **static** and **templates** are corresponding website style and html's mainly used for design.

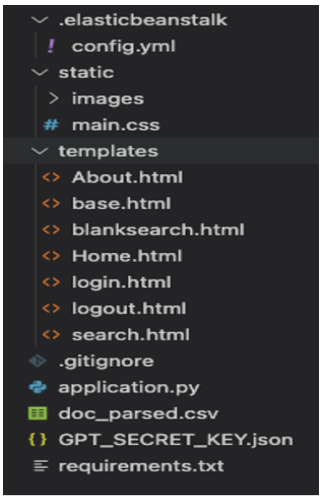


Figure 40: File structure of proposed Web application which will work as MVP.

Here **doc_parsed.csv** is being used as dataset, this dataset contains parsed information of National Instrument 81-102,106 and 107. A glimpse of dataset is given below which contains *document_name*, *document_type*, *document_code*, *document_source*, *part, section, content.*

Important thing to note that GPT-3 can only search through 200 documents in our scenario we can say it can search through 200 sections. For more than 200 sections our program is capable to pass batches of 200 sections and get the top ranked result in the end.

document_name	document_type	document_c	document_s	part	section	content
National Instrument	Investment Funds	81-102	January 2,20	PART 1 DEFINITION	1.1 Definitions	(1) The Canadian securities regulatory authorities in a number
National Instrument	Investment Funds	81-102	January 2,20	PART 1 DEFINITION	1.2 Application	(1) This Instrument applies only to (a) a mutual fund that offer
National Instrument	Investment Funds	81-102	January 2,20	PART 1 DEFINITION	1.3 Interpretation	(1) Each section, part, class or series of a class of securities of
National Instrument	Investment Funds	81-102	January 2,20	PART 2 INVESTMEN	2.1 Concentration Res	(1) A mutual fund, other than an alternative mutual fund, must
National Instrument	Investment Funds	81-102	January 2,20	PART 2 INVESTMEN	2.2 Control Restriction	(1) An investment fund must not purchase a security of an issu
National Instrument	Investment Funds	81-102	January 2,20	PART 2 INVESTMEN	2.3 Restrictions Conce	(1) A mutual fund must not do any of the following: (a) purcha
National Instrument	Investment Funds	81-102	January 2,20	PART 2 INVESTMEN	2.4 Restrictions Conce	(1) A mutual fund must not purchase an illiquid asset if, imme
National Instrument	Investment Funds	81-102	January 2,20	PART 2 INVESTMEN	2.5 Investments in Oth	(1) For the purposes of this section, an investment fund is cons

Figure 41: Parsed Data of law document.

Before running this web app make sure you download zip and extract it later create a python virtual environment. This website is made using the Python 3.8 version. To create environment type "**python3 -m venv env**" this will create virtual environment named "**env**", now enter the environment using command "**source env/bin/activate**", now run the application using **python3 application.py.** This will run the application on local server and can be accessed through **http://127.0.0.1:8080/ .**

Figure 42: Lexata login page

Users need to login with provided credentials (Note: - for now the credentials are hard coded) later the credentials can be dynamic using DynamoDB in AWS and the document data can be fetched through S3.

As we can see in the above figure given below it shows all documents and their section level details. Users need to enter in to search box to search the relevant section they can customize their search by selecting the engine type and number of results to be displayed.

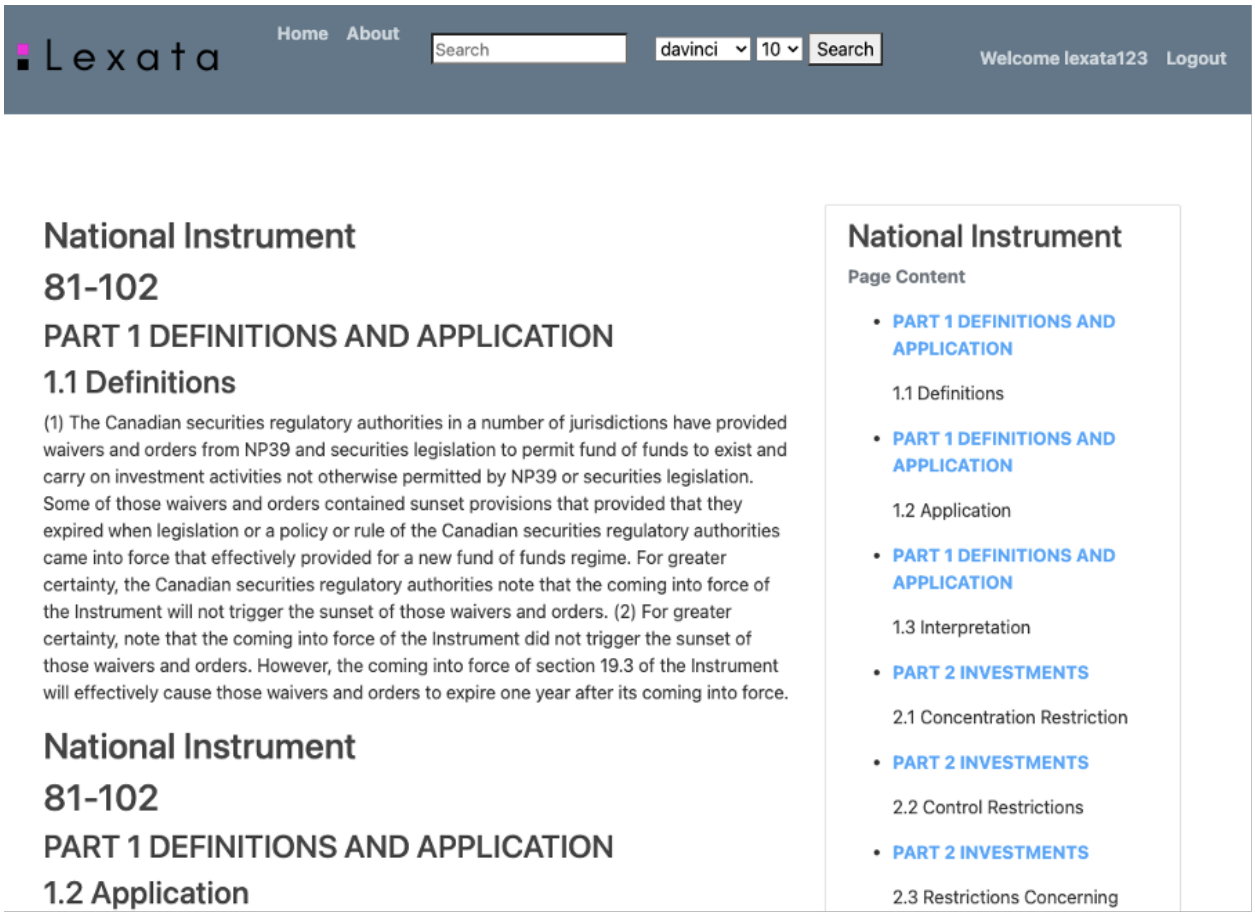


Figure 43: Home page

GPT-3 provides 4 engine supports namely, **davinchi**, **curie**, **babbage** and **ada,** these are different algorithms which score the section in all documents and return the ranked result with highest to lowest score.

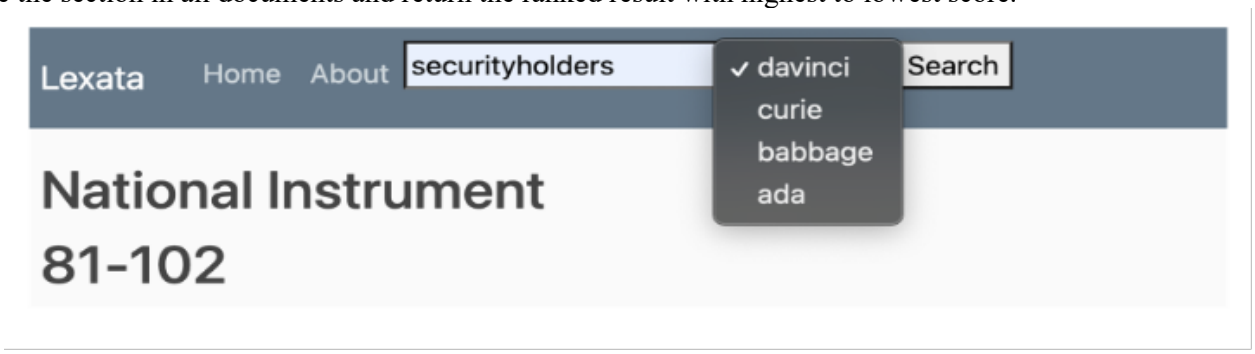


Figure 44: Search engine options

Open AI provides the scores for each section later we choose top results based on our requirement for example we want to look at top 10 documents with high score.

document_name	document_ty	document_ca	document_su	part	section	content	score
National Instrument	81-107	81-107	Unofficial co	National Inst	1.1 Investme	(1) This Instrument applies to an investment	342.419
National Instrument	81-107	81-107	Unofficial co	National Inst	3.13 Fees an	The investment fund must pay from the asse	317.063
National Instrument	Investment F	81-106	Unofficial co	PART 15 ¥¢,(	15.2 Fund of	(1) For the purposes of subparagraph 15.1 (1	295.949
National Instrument	81-106	81-106	Unofficial co	National Inst	7.2 Multiple	(1) An investment fund that has more than c	295.509
National Instrument	81-106	81-106	Unofficial co	National Inst	15.1 Calculat	(1) An investment fund may disclose its mar	288.345
National Instrument	81-107	81-107	Unofficial co	National Inst	3.1 Independ	An investment fund must have an independe	288.109
National Instrument	81-106	81-106	Unofficial co	National Inst	4.2 Filing of	An investment fund, other than an investme	285.615
National Instrument	Investment F	81-106	Unofficial co	PART 15 ¥¢,(	15.1 Calculat	(1) An investment fund may disclose its mar	282.642
National Instrument	81-106	81-106	Unofficial co	National Inst	15.2 Fund of	(1) For the purposes of subparagraph 15.1 (1	277.665
National Instrument	81-106	81-106	Unofficial co	National Inst	4.5 Approval	(1) The board of directors of an investment f	274.24

Figure 45: Search score for each section in sample csv file.

This documentation is focused on the deployment of our flask website on the Amazon AWS Elastic Beanstalk[14].

Step -1 Creating amazon aws account and IAM user.

Minimum requirement is to create a IAM user and assign the following roles. Makes sure you check the service role and make sure Elastic Beanstalk is there[15].
"AmazonEC2FullAccess","AdminiastratorAccess-AWSElasticBeanstalk","AutoScallingFullAccess", "ElasticLoadBalancingFullAccess",

Step -2 Install Elastic Beanstalk CLI. (We will be deploying using CLI method) and install required libraries

Install elastic beanstalk cli[15] "pip install **awsebcli**" will install the elastic beanstalk cli in your environment. Then simply check the version of eb cli mentioned below.

```
chandrakantprajapati@192-168-1-109 Lexata_Flask % eb --version
EB CLI 3.19.4 (Python 3.8.0)
```

Figure 46: Screenshot of elastic beanstalk version

Open the terminal and go to the directory create a python environment just outside the flask website directory (We don't want to deploy environment it's just for our development).
Activate the environment using **"source env/bin/activate".**

```
chandrakantprajapati@192-168-1-109 Lexata % ls
Lexata_Django                   lexata_new.pem
Lexata_Flask                    new_user_credentials.xlsx
env
chandrakantprajapati@192-168-1-109 Lexata % source env/bin/activate
(env) chandrakantprajapati@192-168-1-109 Lexata % ls
Lexata_Django                   lexata_new.pem
Lexata_Flask                    new_user_credentials.xlsx
env
(env) chandrakantprajapati@192-168-1-109 Lexata % cd Lexata_Flask
(env) chandrakantprajapati@192-168-1-109 Lexata_Flask % ls
GPT_SECRET_KEY.json     doc_parsed.csv          static
application.py          requirements.txt        templates
```

Figure 47: Screenshot flask directory.

Run the following command **"python3 -m pip install -r requirements.txt"** this will install the libraries required for our web application.

Now run the application using the command **"python3 application.py"** this way we can verify the website is running on the

Figure 48: Screenshot of flask local server

Now that your app is running on local http://127.0.0.1:8000/ we are ready to deploy it.

We will go through steps given below to deploy it on elastic beanstalk.[15]

You do not need to be on the environment for deployment so basically deactivate the environment using command **deactivate.** Then proceed with the following steps.

- **eb init -p python-3.8 flask-app** to create the application.
- **eb init** for first time it will ask to setup the access key and secret key
- **eb create flask-env** this will create the environment
- **eb open** This will deploy the app and open the deployed link.

http://flask-env.eba-cq8keis6.us-east-1.elasticbeanstalk.com/

It will also create eb built file. **". elasticbeanstalk/config.yml"** in the working directory. This file contains all deployment information.

Once deployment is done, we can make changes as we desire, and we just type the
"eb deploy" after changes.

Limitations

This project is a collaborative work and an effort made in a stipulated time constraint. As per the requirement of the client a Minimum Viable Product capable of performing the semantic analysis on multiple documents was developed and deployed on the AWS cloud platform. The functionality of the product has been rigorously tested and a rhetoric performance has been analysed. Some of the sporadic issues that can hinder the performance of the model and the MVP are discussed in this section. The automatic scripts that are deployed to scan and remove the irregular cardinalities and parse the data cannot be functional if the structure keep changing. The tables and images that are a part of the data in the legal documents cannot be parsed by the automated scripts. The manual intervention is a requirement in the product to alter the functionalities of the scripts according to the structure of the data. The GPT-3 model is highly capable of performing on unseen data as it has been trained on a huge corpus of data including the legal information, however, analysing the outcomes based on the custom requirements is not possible as there is limited information disclosed by Open AI. The engine performances can be monitored based on the relevancy scores but the fine tuning of these engines and the differences in terms of training information is not available to understand the rational reasoning of the relevancy scores. The Web platform is just a prototype to make the MVP user friendly, better web deployments and websites can be made which can be more secure and structured.

Future Scope

To extend the functionalities of this project, a universal standard of publishing legal documents can be practised so that the automation scripts can perform better and yield results without human interactions. An NLP model can be deployed to predict the parts and sections of the documents which can then be passed into the scripts. The scope of GPT-3 model can be increased by uploading more stacked information and checking the performance, although the engines give good performance results on the unseen data, they are not cost effective. The strongest engine can be too costly for multiple searches for any organization; hence, a proposed pipeline may have use of developed NLP models that are coupled with the integrated GPT-3 to reduce the use of very strong engines.

Current web applications can be scaled up to include the Document upload page where admin can upload and train the model. Also, it should contain the view of files which are available on the GPT-3, or the file which can be removed from Open AI if not required. Currently the semantic search is powered by the Open AI's GPT-3 model and in future various other range of models such as Amazon Kendra and Bert model can be implemented to get more control on custom training and results.

Conclusion

As a part of WIL, we were assigned to work with the Lexata Inc and our main task was to automate the already available Lexata website which requires a lot of manual work and some custom regular expression to provide the smart search for users to provide easy access to legal documents. Our task in this project was to fully automate the process from extracting data, of parsing and prettifying the documents to training the GPT-3 model. Using this framework, it should be able to provide users with smart semantic search on multiple legal documents. Data extraction and parsing phase was the initial hurdle of this project and the whole team invested a lot of time in data extraction. Later two groups of teams were working parallely to complete the end to end working MVP. One team was assigned to parse the different groups of legal documents such as British Columbia, Ontario and Exemptive relief, while the second team was working on the research and integration of these parsed documents. Finally, as per the client's requirement we have delivered all the tasks assigned to us and also we have provided them MVP deployed on the AWS service.

Appendix

Roles and Responsibilities

Shubhankar Jahagirdar (s3793593)

- Worked 20-25 hours per week for Lexata project.
- Regular 2-3 meetings per week with RMIT Lexata team members to divide and organize the workflow.
- Researched on appropriate python frameworks to render the parsed and collected information and the search results.
- Research, learn and development of Django framework used to render the information.
- Collaboratively working on making python scripts and to prepare data for modelling.
- Researching methods to integrate the existing data from the Lexata website and use that information for modelling.
- Python scripts to prepare automated batches to separate the data and divide it in chunks of 200 sections.
- Working extensively on Web Scrapping techniques to extend the capabilities of the automated scripts to extract the information from the website links.
- Research on various language learning models and published papers on Open AI platform and working models on semantic searching.
- Working collaboratively as per the expectation of the client on the Open AI and GPT-3 integration with Chandrakant.
- Working on the model development and critical analysis to analyse the effects of the engine performance.
- Single document base model and multi document search.
- Worked on documentation and presentation.

Rafeed Sultaan – s3763175

- Worked 20-25 hours per week for Lexata Incorporated.
- Regular 2-3 meetings per week with the RMIT Lexata Team members to divide and organize the workflow pipeline.
- Research on Legalese Text and identifying the structure of Canadian Securities Legalese Text
- Research on how to do Font-Level and Font-Style Parsing from unstructured PDF documents to create structured tabular dataset using PDF-Miner library in python tool
- Research on the potential of OPEN-AI GPT 3 in making semantic searches
- Experimenting with UI Path software to download all the documents from British Columbia Securities automatically and realizing the website crashes for mass downloads
- Data Collection of pdf documents manually by downloading roughly 500 documents from the British Columbia Securities Commission Website which include Forms, Documents and Appendixes, when only around 20 pdf documents were needed to create the Minimum Viable Product.
- Figuring out the semi-structured pattern in Companion Policy, Multi-lateral Instrument, National Instrument, National Policy Document and Rules and the challenges in
- Automated Script for Parsing Companion Policy, Multi-lateral Instrument, National Instrument, National Policy Document and Rules.
- Automated Script for Parsing British Columbia and British Columbia Documents
- Automated Script for Parsing CSA Staff Notice Documents and figuring out the limitations of Parsing the documents automatically
- Parsed 369 Documents with around 40 pages each for the British Columbia Securities Dataset.
- Generation of British Columbia Securities Law Dataset with around 2195 rows and 8 Columns.
- Report on the issues with the British Columbia Securities Documents.
- Pushing all work to the GitHub Repository.

- Taking Feedback from the Client and improve all deliverables.
- Delivery of all Code and Documentation in the Lexata Git Hub Repository under the "Parsing" Folder.
- Manual Parsing of Multiple Types of British Columbia Securities Documents with Images and Banners
- Research on how to do Web-Scraping using Beautiful Soup Library
- Automated Web Scraping Script to generate the Ontario Exemptive Relief Dataset with which was not needed for the Minimum Viable Product
- Data Cleaning of Ontario Exemptive Relief Dataset to create a standardized dataset removing irrelevant information and non-ascii symbols and characters
- Generation of Ontario Exemptive Relief Document with around 8027 rows and 8 Columns.

Jeevitha Narayanaswamy- S3776688

- Worked 20-25 hours per week for Lexata Incorporated.
- Regular 2-3 meetings per week with the RMIT Lexata Team members to divide and organize the workflow pipeline.
- Selection of project or preferences list given.
- Introduction to the client. (11/03/2021)
- Explanation on the requirements and deliverables.
- Link to first data set collection (BC website) was given and started EDA on the data type for collection.
- Deciding on how to collect data and process it to be used in GPT-3 API- Researching on the various method of input to Open API algorithm.
- Setting up common platform for easy communication (Git hub and Slack communication were established.)
- Researching on the GPT-3 and other semantic search algorithm.
- Mid- Sem Break- 1. Second data set list was given (Ontario Specific dataset).
  2. Deciding on data collection and data parsing- started EDA on the types of Pdfs to be collected and parsed.
- Researching on automation of extracting words from pdf to a data frame or csv.
- Data collection and data parsing started after dividing types of pdfs into Type-1 and Type- 2
- Developing script on text font and text style extraction in python for data collection and parsing into csv.
- Developing the python script using PyMuPDF for text font and style extraction along with cleaning the text using regex.
- Converting pdf data to csv and few URL details were given by the client.
- Appending all the extracted and cleaned data set.
- working on final draft of Ontario specific URL data parsing submitted to client for verification.
- Based on the client's recommendation the changes to the csv were incorporated.
- Working on data cleaning changes as suggested by client to match BC website data and uploading all the scripts to GitHub.
- Final cleaned data csv with parsed data and started working on documentation of the steps to data collection and extraction.
- working on the final report.
- In week 14- working on the final report and product presentation.

Chandrakant Prajapati (s3797785).

- Worked 20-25 hours per week for Lexata Inc.
- Regular 2-3 meetings per week with RMIT Lexata team members to divide and organize the workflow.
- Requirement gathering and suggesting the ideas to implement solutions.
- Assigned to capture the data on the web platform using web scraping python script

- Working extensively on Web Scraping techniques to extend the capabilities of the automated scripts to extract the information from the website links.
- Research study and testing functionality of GPT-3 as per the requirements of the client using the python as the primary open-source language.
- Data preparation for custom training of legal documents on the GPT-3 model and methods to manage the training data set so that it can be scaled further.
- Extensively exploring techniques to implement GPT-3 on our legal documents using python scripting.
- Critical analysis on the performed search queries to understand various engines in the GPT-3.
- Working collaboratively with Shubhankar to develop a pipeline on training single and multi-document semantic searching techniques.
- Integrating the GPT-3 model with Django framework and creating automated functions to process the search query.
- Using authentication techniques to verify the registered users.
- Making a web application for user interaction by deploying the proposed MVP on AWS platform.

Team Activities and collaboration platforms involved.

We spent an average of 20-25 hrs working alone and 4 hours working in a virtual environment collaboratively. We had 4 team meetings of 1 hour each on MS Teams on Monday Wednesday, Saturday and Sunday. In those meetings, we discussed our work divisions. We asked ourselves "How much work was done?", "What obstacles did you face?" and suggested possible "solutions". We had one-on-one meetings with team members whose works got stuck, for example, bugs in their code to resolve their issues at the earliest. We asked them to clarify what the other team-members needed from us to let them continue their work to provide a solution at the earliest. We asked if there was a part that they were stuck on that we might have previously encountered. Finally, we suggested solutions that almost always worked. We used Slack to communicate with their client if there were changes in requirement. Trello was used to divide our work into sprints. Lastly, we used Github to push all our code, data and documentation so that it could be reviewed by the Team Member and Lexata.

Self-Reflection

Rafeed Sultaan-s3763175

During this project, I had to learn Web scraping using Python to extract data from the Web. Previously, I was not an expert in Regular Expressions but that was one of things I had to gain expertise over to parse PDF documents. Furthermore, I got to research on Open-AI GPT3 and the potential of OPEN-AI GPT 3 on making semantic searches. While extracting data from the website I tried multiple methods, but the BeautifulSoup Library was the most useful out of them. It preserves the formatting of the documents. Moreover, it could extract information contained inside HTML elements like '<p>' i.e., paragraphs and ' ' elements. As a by-product of the project, I had to learn how to manipulate and access HTML DOM elements using the BeautifulSoup Library. Moreover, I learned a software called 'UI Path' which can mass download files from any website. This works by specifying the steps and actions in the software. However, the British Columbia Securities website crashes when I try to mass download files.

Exhaustively experimenting with online PDF extraction tools and python libraries, I got an understanding that python has a provision to customize the extraction of data from PDF document. Amongst the python pdf extraction libraries, PDFMINER library breaks the formatting of the documents. Surprisingly, the PYPDF2 library preserves the actual flow of text stored within these documents making it the best choice for extracting data from PDF documents.

Additionally, I had to understand certain patterns in the British Columbia Securities Documents to parse 'pdf' documents from the British Columbia Securities Website document which was extremely time consuming. Eventually, I ended up figuring out the meta-tags for these documents and I tried finding if these meta-tags are common in all documents. Surprisingly, the meta-tags were common in all documents. However, in the future, I would like to organize my time on research instead of only implementation because it caused me to redo the

implementations of the script multiple times. I realized that tokenization using unique tokens like 'PART' ,'BACKGROUND' and other tokens was helpful in dividing the documents into parts and sections. I tried doing font-level parsing, which seemed like a great idea, on British Columbia Securities Dataset as a general template to extract data from the pdf documents. However, the variation in the documents in terms of font, font-style and font-size showed that it was not that fruitful. From studying these documents extensively, I concluded that these type of legalese documents usually follow no standards of formatting. The knowledge that I gained from the use of regular expression helped in removing irrelevant information from the content columns. Using regular expression and tokenization I could fully automate the process parsing the pdf document 'part' – level wise.

One of the challenging that I encountered was that I realized there was no pattern or unique token names in these pdf documents to extract the 'section' within these documents, since these documents were not fully structured. As an additional work I performed the section level parsing manually by separating the section numbers and the section contents in the final dataset. As part of future work, I would like to use a machine learning model to identify the sections in each document. This would considerably reduce the amount of manual effort to separate the section inside the part contents.

Some of the document types like 'CSA staff Notice' had no structure or unique tokens to divide the documents into parts and sections. Moreover, these documents had tables and images which could not be extracted using python scripting. I had to use a bit of manual data entry for CSA documents to merge all the datasets into a unified British Securities Law dataset. The core lesson I learned was that there was no fully automated solution or script to parse and clean all types of documents. Some of the templates will work if there is a pattern. And if there is no pattern or images, we need human-level interaction to identify how the data should be divide into a structured format. As a way to further process the images and tables in the future, I have included a column called 'Flag' to mark sections with tables and images. Lexata said that the table and images are not that important in their process to create a minimum viable product. But I have created the provision to later process rows of the dataset which contained tables and figures. In short, we can conclude that parsing law documents from website pages was significantly easy because the documents are structured but it becomes very time consuming to parse pdf documents as a large portion of these documents have no structure. Sometimes small amount of human-level interaction and automation is faster than fully automated script using state of the art Image Recognition software to solve a simple task like identifying the information (i.e., document name, document type and document code) inside a banner like the 'CSA Staff Notice documents. This project had been a great learning experience, since I got to learn about how a real data science project works from end-to-end in deploying a minimum viable product. Through my painstaking experience of this project, I got to understand that the saying "*80% of the data science process is spent on Data Collection and Data Preparation" is true since it was the most time consuming process.

Chandrakant Prajapati (s3797785)

During my time in Lexata I have worked on various technologies to fulfill the client requirements. It was a great experience to work directly under the CEO and CTO of Lexata Inc. along with the fellow students of RMIT.

In the initial phase of our project, I was assigned to work on the research of the GPT-3 engine and how it can be implemented to work with the legal documents. The Lexata website which is currently running is developed on the WordPress platform with limitations as the document search was manually hardcoded.

I have provided proof of concept that GPT-3 can be implemented and automated to provide smart semantic search. I have also participated and helped my team members to understand the workflow and steps which should be taken to implement the fully automated document search from the parsing phase to searching using GPT-3.

I have also got the opportunity to work on the data extraction and parsing from the web using python scripts which was very helpful for the team who were working on the data parsing phase. This also gave me an in depth understanding of the data and how it can be linked to GPT-3.

After finishing the sample phase of GPT-3 using the sample document while working jointly with Shubhankar we have developed the pipeline to train the documents and also linking the parsed data set to the Django web application which was a proposed replica of Lexata website.

It was challenging for us to create a process flow which can be trained on the multiple documents and a user can query in those documents as OpenAI searches on the 200 documents at one time. We found a way to deal with such a situation by passing the batches of 200 documents and choosing the documents with high scores. Later we combined all top scores and selected the highest scores to decide the winner documents with high probability of query.

After this phase, I have converted the whole script into the python function and linked with the Django website containing similar documents on which Gpt-3 has been trained. Users can search using various engines and also, they can select the top search count as per their convenience. As we were getting promising results in the user search query and critically analysing the performance of engines, I was assigned to deploy the working MVP (Web Application) on AWS.

One of the most important things I have learned is to propose solutions that are easily understandable by an audience who may not have the technological background. It is one of the most important skills as a Data Scientist since every client may have a different background and it is our responsibility to make them understand how things can be implemented based on their requirements.

I have also got the opportunity to work on web scraping and parsing using python. This is also a very valuable learning experience as we need to know the ways data can be enriched and it was fruitful to have this skill.

I have also learned how to design and develop a sample web application that can be integrated with the ML models which enables me to deliver MVPs more easily. Not only that, but also deploying the website on AWS. Sometimes clients may not have the fully functioning python environment. It is one of the best ways to offer a global web application link to test their requirements and assess the implementation very easily.

After fully analysing the GPT-3 and its various engine performance we concluded that it is better than the manual regular expressions which were already there on Lexata website. Also, it was impressive in terms of semantic search as GPT-3 was able to rank the documents even for the normal English language which may not be directly linked to the legal documents. There is a huge range of methods which can be implemented and tested on the legal documents. The good thing is GPT-3 is providing very competitive results and can be used as a benchmark for other models such as Amazon Kendra and Bert model, they are also very famous NLP models which can be incorporated in the future.

Initially as required the MVP to be implemented on the Django Framework and supposed to be deployed on Amazon EC2 instance. After spending a lot of time, I was facing issues with the Nginx and Gunicorn linkage with the Django framework, not only that there were also some issues with the static CSS of Django and the whole process was taking a lot of time. Since our aim was to link Open AI's GPT-3 to work on Legal documents, it should be implemented on the simple web application. I have proposed an alternate method to replicate the similar situation and finally recreated everything into the Flask website and later deployed the website on the Amazon Elastic Beanstalk. I found this method much easier and provided the whole documentation for future changes.

Shubhankar Jahagirdar – s3793593

In the due course of this project, I worked on many new technologies. I got an exposure to work with industry experts, stakeholder, and technical mentors. Understanding and implementing all the requirements and implementing them in an iterative process. The technologies that I worked on for this project include the python Django framework to integrate and deploy the application and the searching results. The open AI GPT-3 pre-trained model and integrate the same with the python framework. Studying and analysing the significance of the GPT-3 and the response returned.

I got an exposure to work on the data collection and parsing of the legal documents. As the information provided existed in the form of PDF and Web links, Web scrapping was a technology that could effectively resolve and handle this requirement. I worked on the beautiful soup library of python and wrote a script that could automate the process of collection of the document from the hyperlinks, parse and store them is the CSV format as per the requirements of the client. Then I worked on gathering this information and using python to prepare the data and store it in dictionaries in order use it to train the GPT-3. I found that the limitations of Open AI in training the engines with not

more than 200 documents, hence, to address this requirement, an automated python script that could take the whole chunk of information and cut it into the batches of 200 documents. This gave me an understanding that the data must be integrated and converted as per the requirement of the ML model. Further, in upcoming iterations as per the requirements of the client the hosted SQL data containing HTML tags were used to populate and train the GPT-3, this took efforts in scrapping the tags and encoding the data without any binary ascii values.

In the due course of development of the MVP, I tried implementing the developed framework on the existing Lexata word-press web-application, however, it failed as I had less knowledge about the word-press applications, and due to the time constraints, those requirements were altered to deploy on the AWS framework. Some of the other language models such as Roberta and Bert were researched, however, the literature survey suggested that the GPT-3 was the best model to be used for the sematic searching techniques as this was trained on the highest number of parameters. In the deployment phase the Django framework was converted into flask due to errors in the deployment, however the work done was templated and can be used on future scope.

This project was a good exposure on the clustering techniques and understanding the GPT-3 models learning capabilities for the synonyms at each forward pass. This made the CNN implementations and transfer learning approaches a confident move in upcoming reflection for the work done in this project. Hence, this project served as a major turning point right from the information gathering to delivering the required results.

Jeevitha Narayanaswamy– s3776688

The Lexata project helped me understanding the importance of data collection and data cleaning in any data science project. The major challenge at my part of work was on deciding how the data can be parsed and cleaned with the help of automated script. After trial and error of various packages and libraries in python PyMuPDF for the pdf extraction was of great help.

Another challenge was to divide the pdfs based on format which took me a while to figure out and with the help of client's domain knowledge the pdfs were divided into Rules, Policies, Instruments and Forms. Cleaning the data took most of the time as the raw data had Nan and other unwanted data. The script failed when an image is encountered, and such pdfs are manually extracted and cleaned which was a major challenge.

The necessary stages in any project are teamwork and communication among the team which was a prominent part in case of our team. Each one of us were able to exchange the ideas and share workload and support each other. The Data Science Post Graduate Project made me understand the importance of professionality, teamwork, time management and transmission of academic knowledge into work related knowledge.

Bonus or Knowledge Acquired:

a. I was able to learn more on usage of regex library in python.

b. Detailed study and trial on PyMuPDF, PDFMiner, PyPDF2, Tabula-py packages and its library for pdf extraction. But was able to succeed with PyMuPDF than other packages.

c. Research on the GPT-3 algorithm helped me gain more knowledge along with Hands on Natural Language Processing.

d. Research on Extracting Images or pictures from a pdf helped me try more packages and user defined algorithm in python.

References

[1] "Leslie McCallum", Linkedin, 2021. [Online]. Available: https://www.linkedin.com/in/leslie-mccallum-lexata/?originalSubdomain=ca. [Accessed: 13- Jun- 2021].

[2] Lexata.ca. 2021. Lexata Inc.. [online] Available at: <https://lexata.ca/lexata/> [Accessed 13 June 2021].

[3] "Ontario Securities Commission Website with search query "exemptive relief"", OSC, 2021. [Online].Available:https://www.osc.ca/en/securities-law/ordersrulingsdecisions?keyword=%22exemptive%20relief%22&field_ord_related_to=All&field_ord_category=5516&date%5Bmin%5D=&date%5Bmax%5D=&sort_bef_combine=field_publication_date_DESC&sort_by=field_publication_date&sort_order=DESC&page=0. [Accessed: 13- Jun- 2021].

[4] "Instruments & Policies", *Bcsc.bc.ca*, 2021. [Online]. Available: https://www.bcsc.bc.ca/securities-law/law-and-policy/instruments-and-policies. [Accessed: 13- Jun- 2021].

[5] "Extracting headers and paragraphs from pdf using PyMuPDF", Medium, 2021. [Online]. Available: https://towardsdatascience.com/extracting-headers-and-paragraphs-from-pdf-using-pymupdf-676e8421c467. [Accessed: 14- Jun- 2021].

[6] "OpenAI API," Openai.com. [Online]. Available: https://beta.openai.com/docs. [Accessed: 01-Jun-2021].

[7] T. Brown et al., "Language Models are Few-Shot Learners", *arXiv.org*, 2021. [Online]. Available: https://arxiv.org/abs/2005.14165v4. [Accessed: 14- Jun- 2021].

[8] T. Magubane, 2021.[Online]. Available: https://www.legalbusinessworld.com/post/the-possible-implications-of-gpt-3-to-the-business-of-law. [Accessed: 14- Jun- 2021].

[9] Y. Jiang and M. Yang, "Semantic Search Exploiting Formal Concept Analysis, Rough Sets, and Wikipedia", *International Journal on Semantic Web and Information Systems*, vol. 14, no. 3, pp. 99-119, 2018. Available: 10.4018/ijswis.2018070105.

[10] "What is Semantic Search? | Ontotext Fundamentals", *Ontotext*, 2021. [Online]. Available: https://www.ontotext.com/knowledgehub/fundamentals/what-is-semantic-search/. [Accessed: 14- Jun- 2021].

[11] Open AI Search Engine, 2021. [Online]. Available: https://beta.openai.com/docs/engines:. [Accessed: 14- Jun- 2021].

[12]"The Ultimate Guide to OpenAI's GPT-3 Language Model," Twilio Blog. https://www.twilio.com/blog/ultimate-guide-openai-gpt-3-language-model [Accessed Jun. 01, 2021].

[13] "OpenAI API," Openai.com. [Online]. Available: https://beta.openai.com/docs. [Accessed: 01-Jun-2021].

[14] "Deploying a Flask application to Elastic Beanstalk - AWS Elastic Beanstalk", Docs.aws.amazon.com, 2021. [Online]. Available: https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-flask.html. [Accessed: 14- Jun- 2021].

[15] "Deploying a Flask App to AWS Elastic Beanstalk", Medium, 2021. [Online]. Available: https://medium.com/analytics-vidhya/deploying-a-flask-app-to-aws-elastic-beanstalk-f320033fda3c. [Accessed: 14- Jun- 2021].