# Swissport Clause/Condition Extraction and Processing

Contents

[Swissport Clause/Condition Extraction and Processing 1](#_Toc156318297)

[Summary 1](#_Toc156318298)

[High-level process 1](#_Toc156318299)

[FAQ 2](#_Toc156318300)

[CUAD v1 experiments 3](#_Toc156318301)

[Summary of the data 3](#_Toc156318302)

[ML/LLM development requirements based on high-level process 5](#_Toc156318303)

[Dataset preparation methodology 5](#_Toc156318304)

[Section importance classifier 6](#_Toc156318305)

[Methodology 6](#_Toc156318306)

[Experimental results on CUAD v1 6](#_Toc156318307)

[Clause identification classifier 6](#_Toc156318308)

[Methodology 7](#_Toc156318309)

[Experimental results on CUAD v1 7](#_Toc156318310)

[LLM validation classifier 7](#_Toc156318311)

[Methodology 7](#_Toc156318312)

[Experimental results on CUAD v1 8](#_Toc156318313)

[Variable extraction LLM 8](#_Toc156318314)

[Methodology 8](#_Toc156318315)

[Risks 8](#_Toc156318316)

## Summary

Blah

## High-level process

1. Documents are extracted via OCR process (extraction pipeline).
2. Text output from extraction process is split into sections (new line separators, full stop separators).
3. A language classifier classifies each section as to its importance (i.e. does it contain information relevant to processing?).
4. Of the important sections, another language classifier classifies the section into one of a set of classes corresponding to the nature of the clause (e.g. cancellation, delay, frequency dependent pricing, etc). The top three highest probability classes are selected.
5. Classified sections are then validated via an LLM which confirms whether the given section statement matches the qualities of the class provided, for each top three classes assigned.
6. The remaining sections that passed the validation step are then sent to downstream LLM functions which extract the relevant variables from the excerpt (if applicable), saving the clause and variables to the database.
7. During billing processing, the system will check for the presence of any relevant clauses to the specific section of the billing pipeline for the customer being processed at the time. If it is present, the variables are extracted and the condition is actioned.

### FAQ

***Why step 3 (section importance classifier)?***

*A null class needs consideration for when clauses do not fit any of the clauses we are seeking. A null class can be applied in step 4, however the intention is to reduce the complexity on this step by filtering irrelevant data first and making step 4 easier to train. The downside is that adding new clauses to the system requires training both step 3 and 4 models.*

***Why step 5 (LLM validation step)?***

*This step might not be necessary, however might help flag misclassifications from the smaller model in previous step.*

***Why not use LLM for all classification steps?***

*The number of classes would require a very large and complicated prompt. The larger the input into an LLM, the higher risk of ‘attention’ issues and misclassifications and/or bad outputs. As the LLM will not be trained specifically towards this task, it would require ‘few-shot learning’ examples in the prompt that would overload the LLM and cause performance issues. Additionally, an LLM can often be ‘too smart’ and make assumptions or extrapolations to produce a result.*

***Why encode and extract variables related to conditions and not pass these to an LLM for the billing processing in step 7?***

*We cannot rely on the wording in the contracts to be unambiguous in regards to how it instructs the reader to apply said clauses. There is often additional context required for full understanding on how to implement a clause which would not be scalable to instruct the LLM at billing processing stage.*

## CUAD v1 experiments

Some of the methods described in the high-level process have been tested and reported using an open-source dataset. The **C**ontract **U**nderstanding **A**tticus **D**ataset (CUAD) is a collection of complex legal documents containing 41 classifications of clauses. The dataset provides both the OCR extracted .txt files of the documents, as well as excel files that contain excerpt to class labelled data. The .txt files contain both the labelled excerpts and ‘irrelevant’ information that isn’t classified (unimportant).

### Summary of the data

* 510 legal documents spanning multiple industries
* 47 categories of important clauses that lawyers look for when reviewing contracts in connection with corporate transactions
* >13,000 labelled excerpt samples

***\*Not all samples are used in experiments.***

#### Type of Contracts and number:

* Affiliate Agreement: 10
* Agency Agreement: 13
* Collaboration/Cooperation Agreement: 26
* Co-Branding Agreement: 22
* Consulting Agreement: 11
* Development Agreement: 29
* Distributor Agreement: 32
* Endorsement Agreement: 24
* Franchise Agreement: 15
* Hosting Agreement: 20
* IP Agreement: 17
* Joint Venture Agreement: 23
* License Agreement: 33
* Maintenance Agreement: 34
* Manufacturing Agreement: 17
* Marketing Agreement: 17
* Non-Compete/No-Solicit/Non-Disparagement Agreement: 3
* Outsourcing Agreement: 18
* Promotion Agreement: 12
* Reseller Agreement: 12
* Service Agreement: 28
* Sponsorship Agreement: 31
* Supply Agreement: 18
* Strategic Alliance Agreement: 32
* Transportation Agreement: 13

#### Class distribution

|  |  |
| --- | --- |
| **Label** | **Count** |
| Parties-Answer | 255 |
| Document Name | 255 |
| Parties | 255 |
| Agreement Date | 231 |
| Agreement Date-Answer | 229 |
| Expiration Date | 225 |
| Governing Law | 222 |
| Effective Date | 209 |
| Effective Date-Answer | 192 |
| Anti-assignment | 188 |
| Expiration Date-Answer | 171 |
| Cap on Liability | 137 |
| License Grant | 128 |
| Exclusivity | 120 |
| Audit Rights | 109 |
| Renewal Term | 95 |
| Termination for Convenience | 91 |
| Renewal Term-Answer | 89 |
| Post-termination Services | 88 |
| Revenue-Profit Sharing | 83 |
| Minimum Commitment | 82 |
| Insurance | 82 |
| Non-Transferable License | 78 |
| Change of Control | 66 |
| Uncapped Liability | 64 |
| IP Ownership Assignment | 62 |
| Notice Period to Terminate Renewal | 57 |
| Notice Period to Terminate Renewal- Answer | 55 |
| Covenant not to Sue | 50 |
| ROFR-ROFO-ROFN | 42 |
| Volume Restriction | 41 |
| Competitive Restriction Exception | 41 |
| Non-Compete | 41 |
| Warranty Duration | 37 |
| Affiliate License-Licensee | 36 |
| Irrevocable or Perpetual License | 31 |
| Liquidated Damages | 30 |
| No-Solicit of Employees | 29 |
| Joint IP Ownership | 23 |
| Non-Disparagement | 19 |
| Third Party Beneficiary | 15 |
| Most Favored Nation | 14 |
| No-Solicit of Customers | 10 |
| Affiliate License-Licensor | 10 |
| Unlimited/All-You-Can-Eat License | 8 |
| Price Restrictions | 7 |
| Source Code Escrow | 6 |

#### Example clause and class

**Text:** “*For purposes of the preceding sentence, and without limiting its generality, any merger, consolidation or reorganization involving Licensee (regardless of whether Licensee is a surviving or disappearing entity) will be deemed to be a transfer of rights, obligations or performance under this Agreement for which Licensor's prior written consent is required. (Page 15)”*

**Label:** Change of Control

## ML/LLM development requirements based on high-level process

* Important/not important section classifier (ML)
* Clause classification classifier (ML)
* LLM validation classifier (prompt engineering)
* Variable extraction LLM (prompt engineering)

## Dataset preparation methodology

Both the ML and LLM models require a robust validation process to accurately assess performance and identify where issues are and how to solve them. Additionally, some of the models require a labelled training dataset for training the models for the task.

Curating a labelled dataset is a time-intensive and laborious task. To reduce this burden, an iterative approach can be implemented where an initial dataset is manually created to train the first iteration of models. These models can then be used to support the data curation process by taking a first pass on new samples requiring labelling, with human validators correcting any mistakes seen.

To generate a dataset of excerpts to manually classify, the full text of the document can be split by new lines and/or full stops – each of these becoming an excerpt requiring classification. ***Other splitting strategies need to be explored as this might not always work!***

An example of what this labelled dataset could look like:

* Source – the document the excerpt is from
* Text – the extracted excerpt
* Importance Category – bool for important / not important
* Clause classification – One class from a set of pre-defined classes
* Var\_n – N many variable columns for each relevant variable requiring extraction. The number of variables (N) is specific to the type of clause.

***Prior to any manual labelling task, an exhaustive map of important clauses and associated variables will be required.***

## Section importance classifier

The objective of this model is to classify sections of a document as either ‘important’ or ‘not important’, where ‘important’ is a clause that needs to be considered in the billing process.

### Methodology

1. Build a dataset of contract excerpts and importance labels. This would be a manual process at first, however once a first iteration model is produced this can help speed up the process via manually reviewing classifications on new (unseen) excerpts.
2. Train an encoder-only small LLM model on the labelled dataset. For example, BERT.
3. Evaluate the model using accuracy metric and confusion matrix to understand performance. Evaluate accuracy at a clause category-level also to identify gaps at a clause category-level.

### Experimental results on CUAD v1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Loss** | **F1** | **ROC AUC** | **Accuracy** | **Training Samples** | **Testing Samples** |
| 0.23 | 96.38% | 96.37% | 96.37% | 4403 | 1376 |

The training immediately had good evaluation metrics by epoch 1 on the validation dataset. It shows that the method is very likely to work for Swissport. The complexity of the task is quite small in comparison to clause identification, and also allows for some class balancing via under-sampling the ‘not important’ class.

## Clause identification classifier

The objective of this model is to classify an important clause into one of a set of clause classes. These classes correspond to different rules that need to be applied in the billing process, such as cancellation clauses, delays, ferry flights, etc. Within this, there are sub-types that require separation, such as cancellation clauses that impact the base rate charges vs cancellation clauses that only charge additional manhours performed.

Due to the sizeable variability of how Swissport may construct a billing method for a given customer, the number of classes/categories could be significantly large and has potential to overlap. This may cause issues for the model in performing its task and may require an additional ‘sub-type’ classifier to be implemented downstream. Alternatively, the LLM validation step could potentially be tailored towards solving misclassification issues.

A made-up example of ‘overlap’ using cancellation clauses:

* Cancellation with notice under a certain threshold reduces the flight cost by x%.
* Cancellation with notice under a certain threshold result in additional manhours performed charged to the customer.
* Cancellation with notice under a certain threshold reduces the flight cost by x%, and any additional manhours performed to the customer are charged.

### Methodology

1. Build a dataset of important contract excerpts and clause category labels. This would be a manual process at first, however once a first iteration model is produced this can help speed up the process via manually reviewing classifications on new (unseen) excerpts.
2. Train an encoder-only small LLM model on the labelled dataset. For example, BERT.
3. Evaluate the model using accuracy metric and confusion matrix to understand performance. In particular:
   1. Misclassifications appearing commonly between 2 or more classes – this is likely to be ‘overlap’.

### Experimental results on CUAD v1

The dataset was initially split into 3,526 samples for training, 694 for in-training validation, and 4,220 held out for testing. The split was at random so as to retain the class distribution. The in-training validation set is used during training to validate the progress of the training the model, terminating training once no progress is made in accuracy metrics (F1 score) – since it is used in training it is excluded from evaluative results to ensure evaluation fairness.

The model used was a pre-trained BERT, fine-tuned on the training and validation data. BERT is an encoder-only small LLM that is noted to have high performance in NLP classification tasks.

At a high-level, the model scores the following:

* **Accuracy:** 74%
* **ROC-AUC:** 87%
* **F1:** 74%
* **Runtime Efficiency:** 62 samples/sec (on a local 12 GB GPU)

Accuracy however is inconsistent across classes. The accuracy by class looks correlated with the number of samples in the dataset. This means minority classes will likely need additional steps to ensure accuracy is consistent:

|  |  |
| --- | --- |
| A green and white bar graph  Description automatically generated | A green and black graph  Description automatically generated |

Observing the confusion matrix reveals some signs of conditions which the model struggles to distinguish between or identify altogether. This would likely be the case for ‘overlap’ between condition wording/intent.

A screenshot of a computer

Description automatically generated

One concern regarding Swissport is the number and diversity of wording of contract clauses, with reflection to the number of samples required for initial training. Training requires a number of samples in order to successfully converge, however manually labelling a dataset takes time, how many samples minimum as a rough estimate do we need initially? What is the class imbalance we will face and how many samples is a minimum?

To estimate this, training dataset was iteratively reduced in intervals of 20% (prior to splitting into train, val sets) with a model trained on each iteration. Accuracy drops with each reduction of samples (as expected); however this is only severely detrimental at roughly 704 samples.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Loss** | **F1** | **ROC AUC** | **Accuracy** | **Training Samples** | **Testing Samples** | **No. Classes** |
| 0.03 | 73.63% | 86.53% | 73.63% | 3526 | 4220 | 47 |
| 0.03 | 72.89% | 86.15% | 72.89% | 2820 | 4220 | 47 |
| 0.04 | 70.26% | 84.81% | 70.26% | 2115 | 4220 | 47 |
| 0.05 | 63.86% | 81.54% | 63.86% | 1410 | 4220 | 47 |
| 0.11 | 19.76% | 59.01% | 19.76% | 704 | 4220 | 47 |

These results being replicated with Swissport data however is down to the distribution of the data and the variance of wording.

## LLM validation classifier

After an initial first pass on classifying the clause category for a given excerpt, the LLM will validate the classification using the top 3 highest scoring classes from the previous classification step. The objective is to reduce the number of misclassifications with a stronger model within a smaller space than if we were to use the LLM to classify from the get-go.

### Methodology

A fabricated example:

Excerpt: *“Where the Carrier has notified the Handling Company in writing of a cancellation, and the time of notification is more than 24 hours before scheduled time of departure, the Carrier will not be charged. If the Carrier informs the Handling Company less than 24 hours and more than 6 hours, 50% of the rates in paragraph 1.1.2 shall apply. If the Carrier informs the Handling Company less than 6 hours, 100% of the rates shall apply.”*

Classification (top three probabilities):

1. Cancellation – 40%
2. Delay – 15%
3. Return to ramp – 5%

Example prompt template:

*“””*

*You are an intelligent classification AI… # system preamble*

*Your task if to classify… # instructions*

*Here are the defining qualities of the categories to choose from:*

*{qualities}*

*Here are the examples for each category to choose from:*

*{examples}*

*Here is the excerpt to classify:*

*{excerpt}*

*“””*

### Experimental results on CUAD v1

Not tested.

## Variable extraction LLM

Once a clause has been classified in regards to intent, it then needs to be encoded in such a way that can be easily integrated into the billing process. A condition that has variables (such as cancellation hours of notification and respective rate reduction) requires said variables to be passed during the billing process step. Therefore, the variables associated with the condition need to be extracted from the contract excerpt.

This step can be performed by an LLM that under instruction will provide the extracted variables as a JSON that can be passed to a database for retrieval during billing.

### Methodology

TBD

## Risks

* Clauses that contain more than one condition
* Clause category overlap making classification tricky
* Clauses that contain varying number of variables (low risk as can be considered fairly easily, just might be time consuming)