Natural Language Understanding and Extraction of Flight Constraints Recorded in Letters of Agreement

Stephen S. B. Clarke[1] and Zhifan Zhu[2]
NASA Ames Research Center, Moffett Field, CA 94034, USA

Olivia He[3]
University of California Santa Barbara, Santa Barbara, CA 93106, USA

Jacqueline A. Almache[4]
University of California San Diego, La Jolla, CA 92093, USA

Krishna Kalyanam[5] and Raj Pai[6]
NASA Ames Research Center, Moffett Field, CA 94034, USA

This paper presents an automated information extraction and inference technique using natural language processing for extracting flight operational procedures and constraints embedded in heritage air traffic management documents. The extracted flight constraints can be digitized and fit into existing airspace information exchange models such as the Aeronautical Information Exchange Model (AIXM). This approach offers a digitized solution to disseminate airspace operating conditions to diverse air users and stakeholders in the National Airspace System (NAS). Furthermore, the digitized flight procedures can provide operational flexibility for emerging advanced air mobility providers and reduce traffic controller workload while maintaining current safety standards. To demonstrate this process, 1,972 Letters of Agreement (LOAs) have been selected for processing, named entity extraction, constraint identification and extraction. This dataset is derived from a subset of documents related to Air Route Traffic Control Centers (ARTCC) operations. We experimented with various traditional information extraction techniques, state-of-the-art machine learning and deep learning models to perform named entity recognition and pattern recognition on our dataset. We present the results from our experiments and demonstrate 99.0% F-1 score for named entity recognition, and a 96.6% accuracy for our entire workflow up to named entity recognition. We also discuss constraint definitions using generic patterned templates and extensions to this work in applying entity linking to digitally extracting relevant constraints.

I. Introduction

The primary focus of this paper is to *identify*, *define*, and *extract* structured *flight constraints* contained within the procedures section of Letters of Agreement (LOAs), and fit them into a standardized exchange model (XM) format e.g., AIXM for distribution to stakeholders in the National Airspace System (NAS). Flight constraints can be defined as a restriction to flights that constrain parameters of flight such as altitude, speed, position, trajectory, heading, and

[1] Software Engineer, Flight Research Associates, and AIAA Student Member
[2] Insert Job Title, Department Name, and AIAA Member Grade (if any) for second author.
[3] NASA Intern and Undergraduate Student.
[4] NASA Intern and Graduate Student.
[5] Sr. Aerospace Research Engineer, Aviation Systems Division and AIAA Sr. member
[6] Senior Technologist, Aeronautics and AIAA Member.

arrival time. A restriction is a defined rule that limits the freedom of flights with specific characteristics. We are interested in digitizing flight constraints and presenting them in a standard format such as the Aeronautical Information Exchange Model (AIXM). AIXM is jointly developed and maintained by the FAA and EUROCONTROL, to provide a globally applicable standard model of aeronautical data and an exchange format to support Aeronautical Information Services (AIS) [1]. Information captured within AIXM include airspace structures, procedures, routes, and flying restrictions. In the current version of AIXM (version 5.1.1), digital Notices to Airmen (NOTAMs) are captured using AIXM and used in select airports throughout the United States. Although AIXM contains definitions for flight constraints and restrictions, there are quite a few types of constraints (in LOAs) that are not defined in the current version of AIXM. We anticipate a future version of AIXM or similar XM model would contain the definitions for all flight constraints of interest to the community.

According to the FAA, LOAs are negotiated "*if the air traffic manager deems it necessary to clarify responsibilities of other persons/facilities/organizations when specific operational/procedural needs require their cooperation and concurrence.*" [2] These agreements are made between different facilities including Air Route Traffic Control Centers (ARTCCs), Air Traffic Control Towers (ATCTs), Flight Service Stations (FSS), approach control facilities, other government agencies, airport managers, aircraft operators, and even commercial space operators [2]. The challenge in extracting information from these documents is interpreting the information from their semi-structured *natural language* form. Luckily, due to recent advances of information extraction in Natural Language Understanding (NLU) [3], relevant tools are becoming increasingly available to tackle this problem. This paper presents the results and progress made in the *first ever successful attempt* in solving this problem using modern NLU techniques.

Constraints found in LOAs can be defined as *required procedures, operations or restrictions imposed on flights, operators, other persons, facilities, and stakeholders* in the NAS. Flight constraints in LOAs pertain to information on procedures or restrictions related to aircraft and flight operations. Some examples of flight constraints include local instrument flight procedures, establishing aircraft call signs and description of airspace areas with special operations [2]. Finding and extracting flight constraints into a digitized form involves a sequence of processes i.e., 1) preprocessing the data, 2) using named entity recognition (NER) to tag the LOA data, 3) identifying and extracting constraint patterns. One of the biggest challenges in this overall process is the natural language data format the flight constraints are written in. These flight constraints are written in different ways, and they need to be generalized into a set of defined classes. Each defined class represents a type of flight constraint, whether it is a local instrument flight procedure or an altitude restriction. A more specific example of this class is provided in section VI.

To achieve the goal of extracting generalized flight constraints, we propose and implement the following steps:

- First, we extract the text and key sections of the document using PDF2Text.
- Second, we use traditional gazetteers[7] and syntactic lexical pattern matching to extract named entities.
- Machine learning is used to assist in flight constraint identification by performing pattern recognition.
- Lastly, pattern templates are generated from the data samples to be fit and extracted from the text.

Throughout these steps, we conduct continuous manual validation (using SMEs) of the intermediate results and refinement of the model development process. Fig. 1 gives an overview of the above methodology and workflow, whereas Fig. 2 an example of data flow throughout this process.

The rest of the paper is organized as follows. Section II contains relevant background to NLU and introduces some methods we explored such as *named entity recognition*, pattern finding using *clustering*, and information representation models. Section III describes the processing of the LOA data through cleaning, preprocessing, and data segmentation. Section IV discusses our custom NER approach using gazetteers and syntactic lexical patterns. Section V discusses the manual validation effort performed, and results from the preprocessing and NER sections. In section VI, we provide some initial findings of constraint pattern recognition and close with some concluding remarks in section VII.

[7]A gazetteer is simply a collection of common entity names typically organized by their entity type.

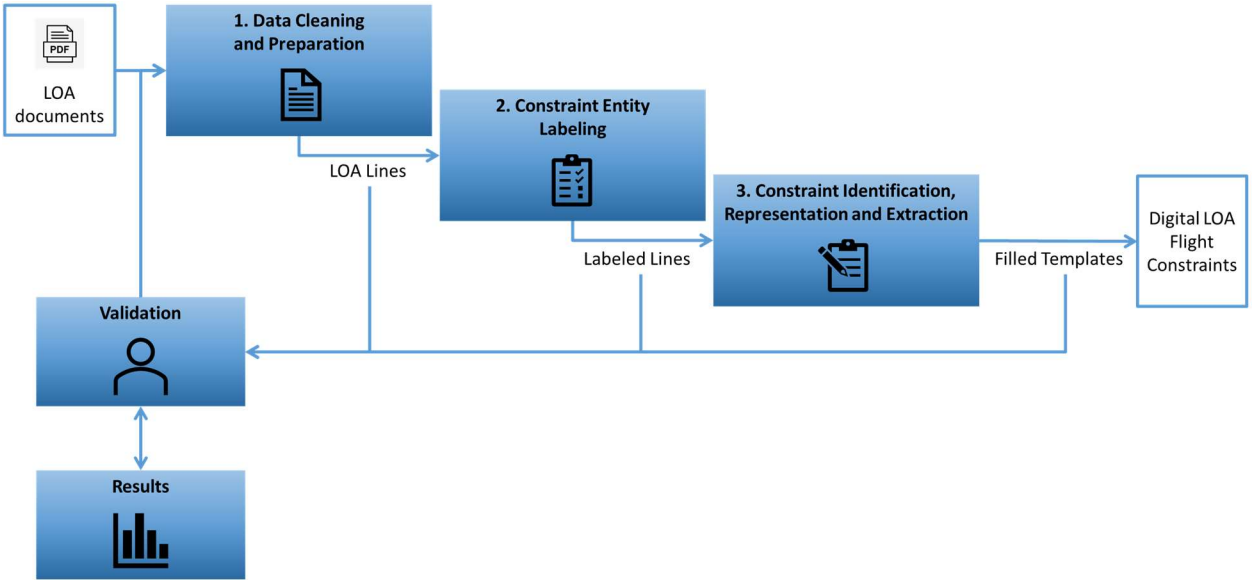


Fig. 1 Development Approach

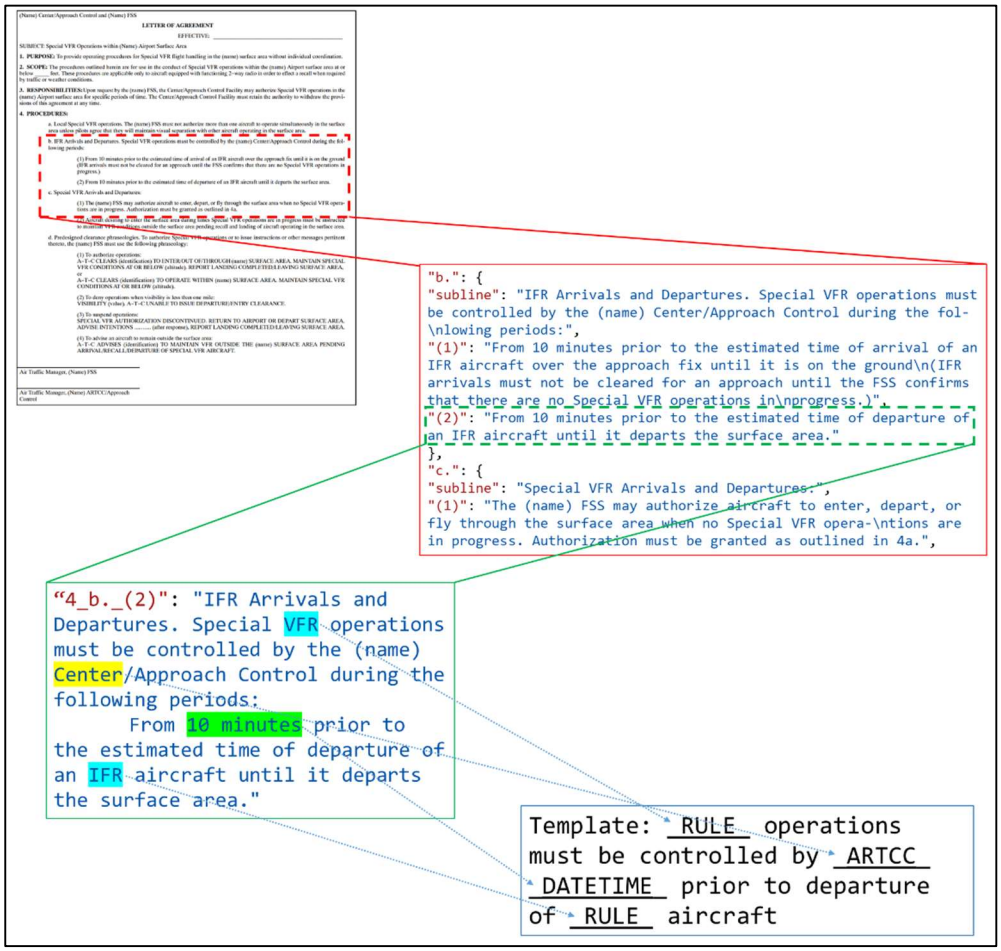


Fig. 2 LOA Information Extraction Process Overview

II. NLU Background

NLU and NLP (Natural Language Processing) have been a hot research topic in recent years, especially due to the explosion in the development and application of data-driven deep learning technologies. As a side benefit, many new methods and techniques have been introduced to tackle domain-specific problems that contain less data but are constrained to information provided in the domain. In 2021, it was estimated that about 80% of all corporate data was contained in unstructured text [4]. This trend is particularly true in the aviation domain, which is why NLU has been employed in interpreting and digitizing legacy text documents. Specific use cases include document anomaly detection, aviation safety report analysis, and sentiment analysis on customer service interactions [4]. This paper looks to expand the list of use cases to heritage ATM documents such as LOAs. Since much of the natural language data in LOAs is human-readable, deploying methods to automate the information extraction and digitization is a key step towards adoption of digital services onboard the aircraft. Therefore, the process we lay out is a key enabler for the FAA's Next Generation Air Transportation System (NextGen) modernization efforts [5]. However, given the rigorous standards set by the FAA, it is imperative that the extracted constraint information accurately represents the written text without any room for misinterpretation. In this work, our FAA partners specified the acceptable performance metrics for the NLP task. They also helped improve our results via a continuous validation (using SMEs) and model development process (shown in Fig. 1).

A. Information Extraction

Information extraction (IE) is the task of converting unstructured data, in our case ATM related text, into structured data that can be easily digestible by machines through search, querying, and data mining [3]. Most of the existing research on IE has been done on domain-specific documents [3]. Therefore, which approach will yield the best performance is highly dependent on the nature of the data source. Some common automated methods cited in the literature for the aviation domain include named entity recognition (NER) [6], pattern finding using clustering [7], and template expression matching [8].

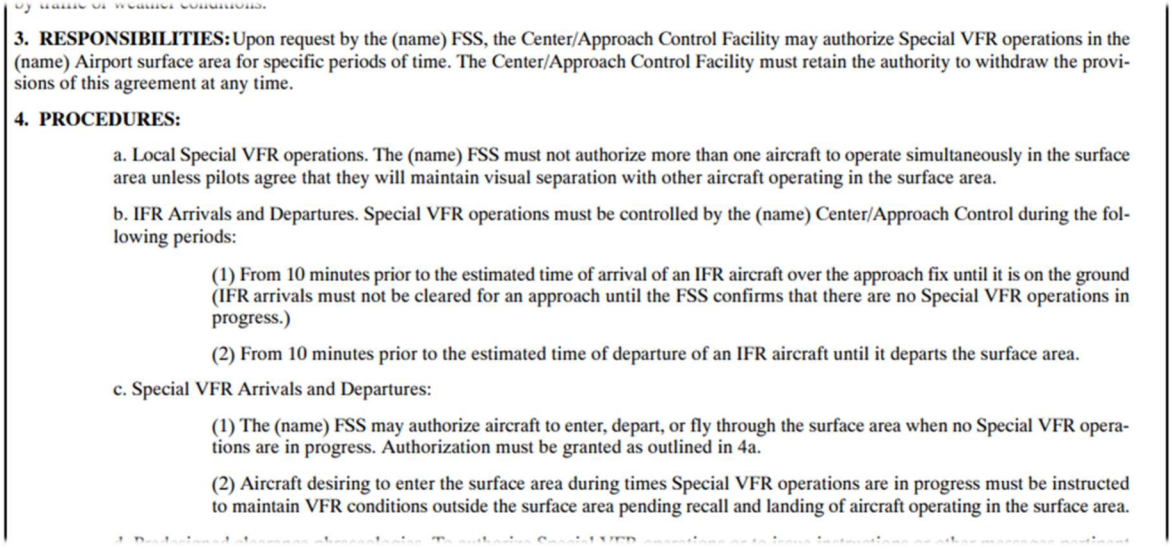
NER classifies unlabeled text according to pre-defined categories by using NLP to automatically tag named entities. These classified categories are typically nouns, or numbers and phrases that represent nouns. There are different models and workflows used today ranging from traditional (rule-based, unsupervised learning, and feature-based supervised learning) systems to more modern deep learning techniques [9]. [8], [10] uses rule-based gazetteers to match tokens in the text with named entities, whereas [6] uses a machine learning approach. Both extraction methods resulted in high-accuracy extraction of named entities. We firmly believe that due to the precise and consistent language unique to aviation literature, rule-based entity extraction is much more promising in contrast to other domains that contain naturalistic prose like clinical notes [11]. This sentiment of domain-specific rule-based matching is also shared in the discussion of [10]. The precise language used in ATM documents is due in part to FAA Order 1000.36 – *FAA Writing Standards*, a document that highlights the use of short and concise language throughout all FAA documents. There is also a great number of standard abbreviations and acronyms used in these texts ranging from facility names to weather patterns.

Clustering has become an increasingly popular NLP technique to uncover hidden patterns and build initial classes to categorize the information within the text. It can be used in tandem with template matching to identify and generate templates before moving onto the extraction process. In [7], clustering was used to identify potential hidden aviation safety anomalies that were previously not classified. Data-driven approaches like clustering can be incredibly useful in understanding unstructured text and can be applied towards identifying flight constraints within LOAs.

Both NER and clustering can be useful in identifying specific classes of text information contained within a document. After identifying common patterns and clusters, logical classes need to be defined which generalize the information found in the data samples and represent the mentioned flight constraint. This step is vital because it is unlikely that techniques like unsupervised clustering will guarantee perfect classes for our use case. Instead, clustering will be used as a tool to assist in the creation of flight constraint classes. When defining the classes, this can be done in different ways including using templates [8], building a knowledge graph [12] or fitting the classes to existing information exchange models like AIXM. A knowledge graph would be a good representation of the flight constraints, but it requires both named entities and entity relationships to be well-defined within the data. An alternative would be fitting the data into an existing information exchange model, but LOAs contain constraints beyond the scope of current operational models like AIXM. Templates appear to have the lowest barrier to entry since they only require named entities to be generalized instead of entities and relationships. Therefore, for this initial pass of defining information contained within the LOAs, we use templates. In future iterations of this work, we may look to create an extension of AIXM that would represent the constraints in a more refined way than templates.

III. Data Cleaning and Preparation

The LOA dataset provided by the FAA contained 1,972 documents pertaining to ARTCC operations stored in a portable document format (pdf). Starting with this data, several steps are followed to convert the raw pdf files into data units that can be manipulated by NLP. This process starts with converting the files into raw text, organized by sections and headings. Then, the document is broken up into data units which are defined as *the smallest contiguous text (of information) that contains a flight constraint*. Fig. 3 shows a template LOA that would be used for a Center Facility or FSS. Most LOAs follow a similar format, but there is a large variety of different information and organization found among the documents. LOAs typically have multiple sections such as *Purpose, Scope, Responsibilities,* and *Procedures*. However, not all documents contain these section titles and ordering. Also, flight constraints are not contained in every section of an LOA. Therefore, it is necessary to down select the documents based on sections that contain flight constraints. These sections generally fall under the Procedures section, but have many names such as*: Procedures, Arrival Procedures, Departure Procedures, General, General Procedures,* etc. For simplicity, all these titles will be referred to as *procedures sections*.



by traffic or weather conditions.

3. RESPONSIBILITIES: Upon request by the (name) FSS, the Center/Approach Control Facility may authorize Special VFR operations in the (name) Airport surface area for specific periods of time. The Center/Approach Control Facility must retain the authority to withdraw the provisions of this agreement at any time.

4. PROCEDURES:

a. Local Special VFR operations. The (name) FSS must not authorize more than one aircraft to operate simultaneously in the surface area unless pilots agree that they will maintain visual separation with other aircraft operating in the surface area.

b. IFR Arrivals and Departures. Special VFR operations must be controlled by the (name) Center/Approach Control during the following periods:

(1) From 10 minutes prior to the estimated time of arrival of an IFR aircraft over the approach fix until it is on the ground (IFR arrivals must not be cleared for an approach until the FSS confirms that there are no Special VFR operations in progress.)

(2) From 10 minutes prior to the estimated time of departure of an IFR aircraft until it departs the surface area.

c. Special VFR Arrivals and Departures:

(1) The (name) FSS may authorize aircraft to enter, depart, or fly through the surface area when no Special VFR operations are in progress. Authorization must be granted as outlined in 4a.

(2) Aircraft desiring to enter the surface area during times Special VFR operations are in progress must be instructed to maintain VFR conditions outside the surface area pending recall and landing of aircraft operating in the surface area.

d. Prolonged absences: Phraseology. To authorize Special VFR operations or to issue instructions or other messages pertinent

Fig. 3 Snippet of LOA for Center Facility/FSS. For a complete LOA, see [2]

B. Document Cleaning

Of the entire dataset, not all 1,972 documents contained relevant information or are able to be parsed into the pipeline. Some of the documents retrieved are not LOAs, but rather FAA facility notices. Some documents do not contain procedures sections, which means they do not contain flight constraints and are removed from the analysis. Lastly, some pdf documents contain multiple LOA documents within one file which requires manual processing to conform with the rest of the data. After these preliminary steps to generate a consistent dataset*, 1,237 LOA documents* are selected for analysis.

C. PDF Document to Section Text

After the document cleaning, the selected LOAs undergo document preprocessing to extract section texts. The document preprocessing consists of PDF to text conversion and section extraction. To convert pdf documents to text, Amazon's *Textract,* a cloud-based API service involving optical character recognition, or OCR, is used. This service takes the pdf documents as inputs and returns a json file which contains the extracted text lines from the document. The text lines are then fed to a customized parser, based on the LOA template format shown in Fig. 4, to extract the content in headings and sections.

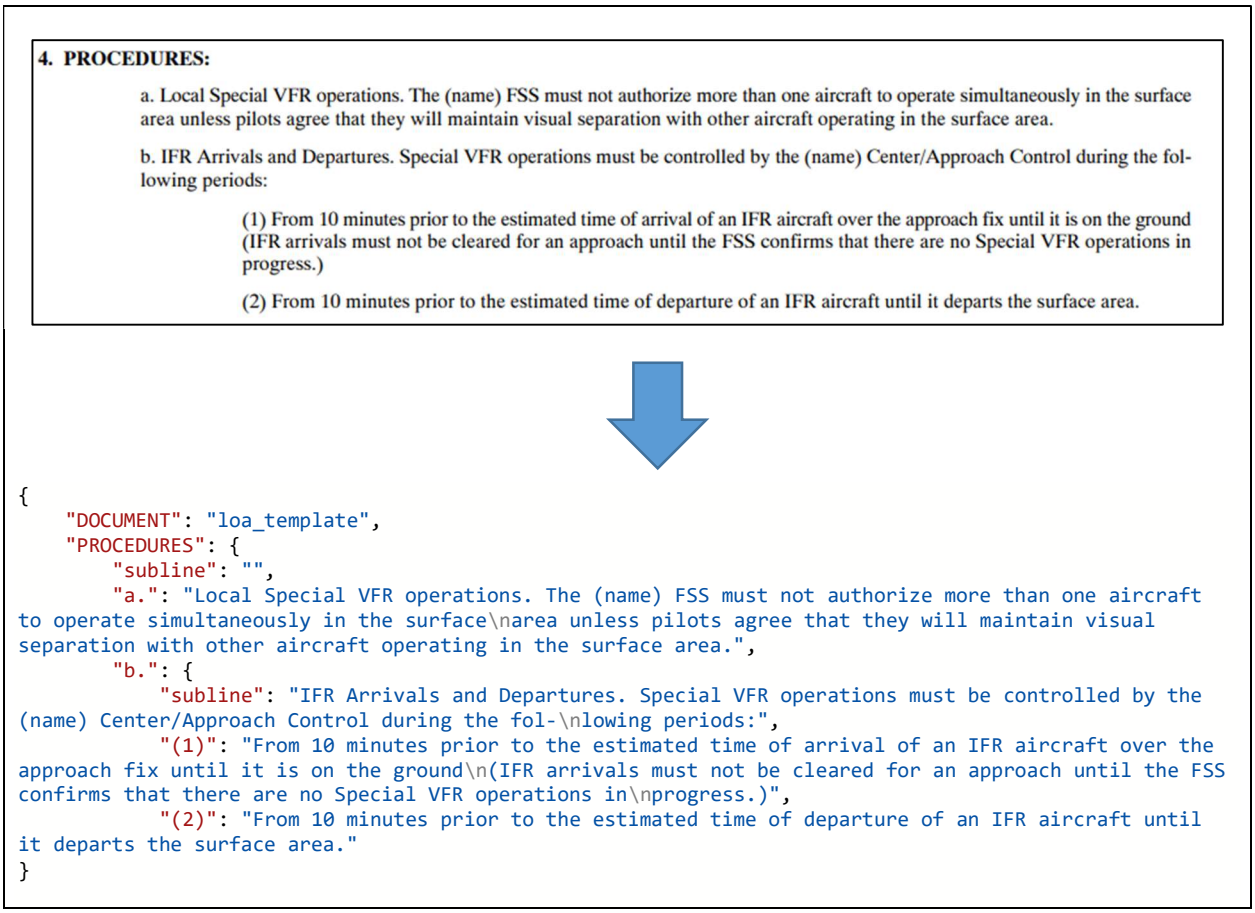


Fig. 4: PDF Document to Section Text Input versus Output[8]

After this process is complete, *1,332 procedures sections* are extracted from the documents and are ready for further analysis. Note that the 1,332 procedures sections extracted are greater in number than the original 1,237 documents because some documents contain more than one procedure section.

D. Unit of Analysis

Now that sectioned text is available, a data unit size must be determined to facilitate identification of flight constraints. A LOA procedure section typically contains multiple constraint statements. Instead of attempting to recognize multiple constraints simultaneously from a whole section text, we break down the section text to smaller pieces with the assumption that a constraint is specified in an independent piece of text, or text unit of analysis. Then the question is: *what unit of analysis would allow us to build a dataset with its data samples (or unit of analysis) having the best chance of holding a single or no constraint?* In other words, data samples that do not contain multiple constraints. There are four choices to consider as shown below.

Entire Procedure Section

The largest unit that can be considered is the entire document. This would include all procedures sections and subsections in one data sample. This size is too big since each data unit would contain multiple constraints without a good way to distinguish between different flight constraints.

Sub Section Level

[8] Note that LOAs are complex documents with different text formats and visual charts. For simplicity, graphic charts and figures are ignored, and table cell relations are not retained in the generated output

This data size would include all sub-sections of each procedures section. Although this narrows down the amount of flight constraints contained within each data unit, there would still be multiple constraints. Another problem with this size is the fact that not all LOAs follow the same formatting. The content of a *sub-section* in one document may be similarly captured in the *section* level of another document.

Line Level
Line level is defined as the deepest sub-section within a document. These data units typically contain at most one to two sentences. The benefit of this over the sub-section level, is that we can consistently define the deepest sub-section of a document.

Sentence Level
LOA lines typically consist of one or more sentences and thus could be split further. However, this unit of analysis likely has dependencies with other sentences within the line. Also, the sentence length depends on the author of the LOA and may contain different amounts of information than what is shown in the template.

After a thorough analysis and review with SMEs, we selected the *line level* option. Through a manual validation process (detailed in Section V), we confirmed that almost all line levels contained either one or zero constraints. Additionally, some further augmentation is done to the *line level* data units depending on what application or NLP task is being performed. Most notably, for manual validation of the data, parent lines are pre-pended to each line level to give context to the user reading the line. Table 1 shows an example of how both types of line units appear. Although the line without pre-pending does not show a complete constraint, it has proven useful in certain NLP tasks like clustering. It is believed that the pre-pending caused problems when clustering because the pre-pended words became a source of similarity between multiple data samples that appeared in the same document, since most of the samples would have the same starting text. Alternatively, with other tasks, the pre-pending has the benefit of giving context to all line levels which means each constraint can be treated independently. As for pre-pending, one could look at the second line in the '*Lines with Pre-pending*' column in Table 1 and understand the context such as which ARTCC is responsible, and the type of aircraft (arrivals)without having to read the first line.

Original Procedure Text	**Lines with Pre-pending**	**Lines without Pre-pending**
5. Arrival Procedures: a. The ARTCC must route arrivals: i. Requesting at or above 12,000 ft MSL via NAVAID unless otherwise coordinated	The ARTCC must route arrivals: Requesting at or above 12,000 ft MSL via NAVAID unless otherwise coordinated	Requesting at or above 12,000 ft MSL via NAVAID unless otherwise coordinated
ii. Requesting below 12,000 ft MSL via STAR unless otherwise coordinated	The ARTCC must route arrivals: Requesting below 12,000 ft MSL via STAR unless otherwise coordinated	Requesting below 12,000 ft MSL via STAR unless otherwise coordinated

Table 1: Line Augmentation Example

Given the 1,332 procedures sections and data unit selection, the final dataset includes *23,911 lines*. Now, the generated dataset is ready for extracting entities and identifying constraints.

IV. Constraint Entity Labeling

To begin the process of identifying and extracting flight constraints within the LOA data, a method is needed to filter and search the dataset for important, domain-specific keywords. The most appropriate NLP technique to solve this problem is NER. NER is typically used for identifying and extracting domain-specific keywords, or entities, from a body of text and classifying them into specific categories [9]. First, a list of entity labels must be identified for extraction. Based on interactions with SMEs, we compiled the following list of entity labels to be used in our work.

- Aerodrome
- Altitude
- ARTCC
- Block Altitude
- Contact
- Datetime
- Distance
- Document
- Equipment
- Heading
- Rule
- Position

- Sector
- Separation
- Speed
- Tower
- TRACON[9]
- Turn

Methods for the extraction part of NER range from traditional approaches such as rule-based and unsupervised machine learning, to more complex and recent methods that use deep learning [9]. Although future work may include comparisons of different NER methods, the method adopted in this work uses gazetteers and rule-based matching. Gazetteers are domain-specific dictionaries which contain long lists of a specific entity [9]. The aviation domain is a great use case for gazetteers since many facility names like *Airports, ARTCCs, TRACONs* and other facility names are well documented and publicly available. Rule-based matching using syntactic-lexical patterns is also an effective method of entity extraction [9] in this domain since labels like *Altitude, Datetime, Distance*, etc. follow a consistent pattern throughout the documents. All the entity labels listed fall into at least one of these extraction methods, while both methods are used in combination for specific cases which contain both common names and patterns. For example, there are commonly known *documents* of interest referenced within the LOA text such as "Standard Operating Procedures," (which can be added to a gazetteer) but also documents that follow a syntactic-lexical pattern such as "FAA Order JO 7210.3," where "7210.3" could be replaced by other numbers.

A. Gazetteers

In our work, to date, gazetteers have been generated for the labels *Aerodrome, ARTCC, ATCT, Document, Equipment, Fix, Heading, Rule,* and *TRACON*. Each individual gazetteer/entity is split into two types: cased and uncased formatting. Naturally, items that appear as cased will be tagged as an entity only if the string matches the proper casing, while those that are uncased will be tagged regardless of case. Airport codes like 'SAN' (San Diego International Airport) should only be tagged as [AERODROME] if they are completely in upper case, otherwise 'San Francisco' would be mistakenly tagged as [AERODROME] Francisco. A limitation to this method is its inability to intelligently handle typos within the document since exact string matches are being identified.

B. Syntactic-Lexical Patterns

Along with specifically defined entities stored inside of the gazetteers, specific string patterns are also developed to match more general patterns. One of the most common patterns are *altitudes*, as demonstrated in Fig. 5. This pattern is made up of two primary sections, a numeric value and unit of measurement. These patterns have been crafted to handle multiple different notations such as 'feet,' or 'ft,' as well as different measurement units like 'mean sea level' (MSL) and 'above ground level' (AGL). Another closely related pattern are *distances*, which have a nearly identical pattern, but the unit of measurement includes keywords like 'nautical mile.'

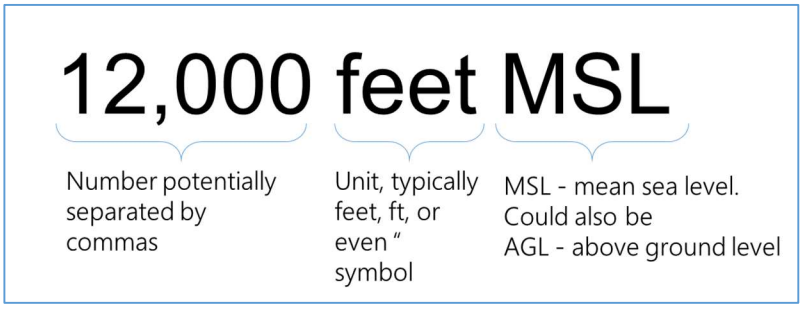


Fig. 5: Altitude Pattern

Although very sparse, certain errors have been found within the labelled dataset that revolve around numeric values without properly labelled units such as in the statement "Aircraft arriving at 12,000." Although a subject matter expert will recognize that the '12,000' refers to an altitude, the rule-based pattern matching does not have the ability to understand the statement as containing an altitude without proper context (in this case units).

C. Tagging

Given the created gazetteers and syntactic-lexical patterns, the LOA dataset is ready to be tagged. Both gazetteers and syntactic-lexical pattern algorithms are first independently evaluated. Entities in the gazetteers and stored string

[9] TRACON: Terminal Radar Approach Control Facility

patterns are searched for throughout each line level and tagged. After both are evaluated, a union function is used to combine the results and handle any overlapping entities between the two methods. This handling function will take the entities that contain the largest number of tokens. Take the reference in Fig. 5 as an example. The first algorithm, using a gazetteer would label the text as "12,000 feet [AERODROME]" because MSL is an acronym for Northwest Alabama Regional Airport. When the second algorithm (syntactic lexical pattern) tags the text, the output would just be [ALTITUDE]. Finally, a union function would determine that since the second algorithm covers more words than the first, the second algorithm's *tag* will be chosen as the final output. In the next section, we lay out the validation process and then analyze the results obtained after applying the chosen NER method to the dataset.

V. Validation and Results

After completing the data pre-processing and analysis steps in the previous section, we deployed a manual validation method which involves checking to see if:

1. The PDF2Text and line splitting are returning correct data samples. These correct data samples follow the definition of 'line level' described earlier.
2. The NER method is properly extracting entities, highlighting defined entities while not highlighting tokens which are not considered in the label definitions.
3. If our assumption is valid that the line levels contain at most one flight constraint.

While this manual effort was underway, it was beneficial to also get an understanding of the distribution of flight constraints versus other (non-flight) constraints, such as procedure constraints for filing flight schedules, within the dataset. Given these conditions, a collaborative workflow, shown in Fig. 6, was set up between researchers developing the method and subject matter experts reviewing the methodology and results. This process was set up to be iterative in nature. Each round of validation would first give an overview of how the developed algorithms are performing, as well as provide feedback to update and refine those algorithms. The process will be repeated until the accuracy of the algorithms hit a specific benchmark. After discussion with FAA collaborators, this benchmark was set at 90% average accuracy for the NLP tasks (preprocessing and NER). As will be discussed in the following sections, this goal was achieved after two rounds of the process.

Fig. 6 Validation Workflow

A. Validation Annotation Setup

Table 2 shows all the information gathered during the validation step. It is split into two categories: *NLP Error*, and *Constraint Type*. NLP errors account for any type of processing error throughout the workflow, from PDF2Text (validation code 1) down to NER errors. A single sample will be tagged with a NER error (validation code 2) if there

is one or more mistakes from the NER algorithm. Also, there is a validation code 3 to combine both line creation and NER errors so that there is no need to assign more than one code per sample.

The constraint types are created to capture the assumption regarding one constraint per data sample, as well as give a distribution of constraint types within the data. Desired flight constraints are considered nominal (validation code 0). Non-flight constraints (validation code 4) are defined as *constraints that do not pertain to pilots or aircraft*. These are typically constraints put on controller-to-controller procedures between facilities. Validation code 5 is used to identify any samples that contain more than one constraint to ensure that our previously mentioned assumptions hold about the data unit size. Lastly, error code 7 was introduced in the second round to identify data samples that contain no relevant or meaningful information. A common example of these is "Facility must:" or "Arrival Procedures:" which appear as a parent line within the *Procedures* sections.

Validation code 6 is used when an annotator is unsure of which code to label the sample as for others to review. This could be for either an NLP error, or constraint type.

Validation Code	NLP Error Definition	Constraint Type Definition
0	No NLP error	Flight Constraint (nominal)
1	Line Creation Error	-
2	NER Error	-
3	NLP Errors 1+2	-
4	-	Non-flight Constraint
5	-	Multiple Constraints
6	Unknown	Unknown
7	-	No constraint information[10]

Table 2: Validation Code Definitions

Along with giving insights to the processes described, this validation effort also led to improvements in the NLP workflow. Recall that a baseline of *90%* NLP accuracy was set as the target goal. This NLP accuracy is an average of both line splitting accuracy and NER accuracy. To improve the accuracy between validation rounds, notes are taken when there is an NLP error, compiled, and used to fix any errors throughout the pipeline. This includes adding or removing entities from the gazetteers or modifying the syntactic-lexical patterns to capture more general forms of entities (illustrated in Feedback step of Fig. 6). This is an iterative process and helped improve the NLP methods accuracy towards the stated goal.

B. Results

This validation process begins with a small dataset of 20 samples that are tagged independently by every annotator (we employed six different annotators). The results are then compared, along with discussion to calibrate the annotators to the task to ensure a higher level of consistency across all annotators. After calibration, 1,000 randomly selected data samples are chosen for a first round. This number is chosen to get a diverse range of examples for algorithm updates if performance is low. The samples are tagged with the above defined validation codes and then totaled for review. Recall that the validation goal is to achieve at or above 90% accuracy for the NLP tasks. The first round, with a total of 88% of NLP accuracy, was very promising but still required a second round of validation to ensure that follow on NLP steps (mainly constraint extraction) is minimally affected by the error in previous steps. For the second round, only 500 samples (different samples from the data used in the first round) were chosen because we were confident that the 90% accuracy mark would be achieved with the algorithm updates gathered from the first round. This proved to be true, with the second round NLP accuracy being 96.6%.

[10] *No constraint* was evaluated only in the second round of validation

Round 1 Examples	Round 2 Examples
Aircraft in the area of V191 and V313, at or below 4,000 feet MSL ALTITUDE , and south of KMTO AERODROME , at or below 5,000 feet MSL ALTITUDE , will normally be transferred non-radar.	Aircraft in the area of V191 and V313, at or below 4,000 feet MSL BLOCKALT , and south HEADING of KMTO AERODROME , at or below 5,000 feet MSL BLOCKALT , will normally be transferred non-radar.
FACSFACSD will coordinate daily time block transits of C1156 at 0700L and 1400L. FACSFACSD will notify Area A Front Line Manager (FLM)/Controller in Charge (CIC AERODROME) time and altitudes available for transit.	FACSFACSD will coordinate daily time block transits of C1156 at 0700L DATETIME and 1400L DATETIME . FACSFACSD will notify Area A Front Line Manager (FLM)/Controller in Charge (CIC) time and altitudes available for transit.
Aircraft filed between the HLN AERODROME VORTAC 107° radial and the HLN AERODROME VORTAC 174° radial from Big Sky Approach when operating and at all other times from ZLC ARTCC sector 06 SECTOR .	Aircraft filed between the HLN VORTAC EQUIPMENT 107° HEADING radial and the HLN VORTAC EQUIPMENT 174° HEADING radial from Big Sky Approach when operating and at all other times from ZLC ARTCC sector 06 SECTOR .
ARTCC must clear arrivals to the non-radar fixes depicted on Annex 1 at 11,000.	ARTCC ARTCC must clear arrivals to the non-radar fixes depicted on Annex 1 at 11,000.

Table 3: NER Examples

Table 3 gives examples of how the NER process tags the data and has been updated between the two rounds of validation. In the top example, one could argue that the first round tagged the ALTITUDE labels correctly. However, given the initial label definitions, a BLOCKALT label fits the examples much better. The second example gives a common mistake, where a general acronym overlaps with a three-letter airport code. To fix this, the acronym 'CIC' was removed from the AERODROME dictionary since it is more likely to be 'Controller in Charge' than 'Chico Municipal Airport.' Finally, the bottom example shows a limitation of the rule-based approach. '11,000' is not captured in either iteration of the models. It was not captured because the number was not followed by a specific unit of measurement even though most subject matter experts would easily identify it as an *altitude* from the context.

These results are investigated in more detail in Fig. 7, which shows an overview of each NLP error type. It is noticed that the line splitting accuracy did not change between rounds one and two, yet there were small changes to the line-splitting algorithm. Although minimal increase was expected, no change still gives confidence that the splitting is done correctly on different data samples. It is likely that the change only affected samples present in the first validation round. Another observation is that the total NLP accuracies are less than the average of the two line-splitting and NER accuracies. This is due to validation code 3 which gets counted as both a line splitting and a NER error.

Fig. 8 gives specific detail on the NER algorithm in terms of recall, precision, and F1-Score which are common metrics to measure NER performance [9]. There appears to be a significant improvement in the NER model between both rounds, and the performance jumped from 94.8% F1-Score to 99.0% F1-Score in round two. This is comparable and slightly better total F1-score, to similar applications using open-domain data [10]. Also, while our method worked well for the targeted label set, it would likely have failed to target broader entities like 'Person' which cannot easily be defined in a dictionary or general pattern [10].

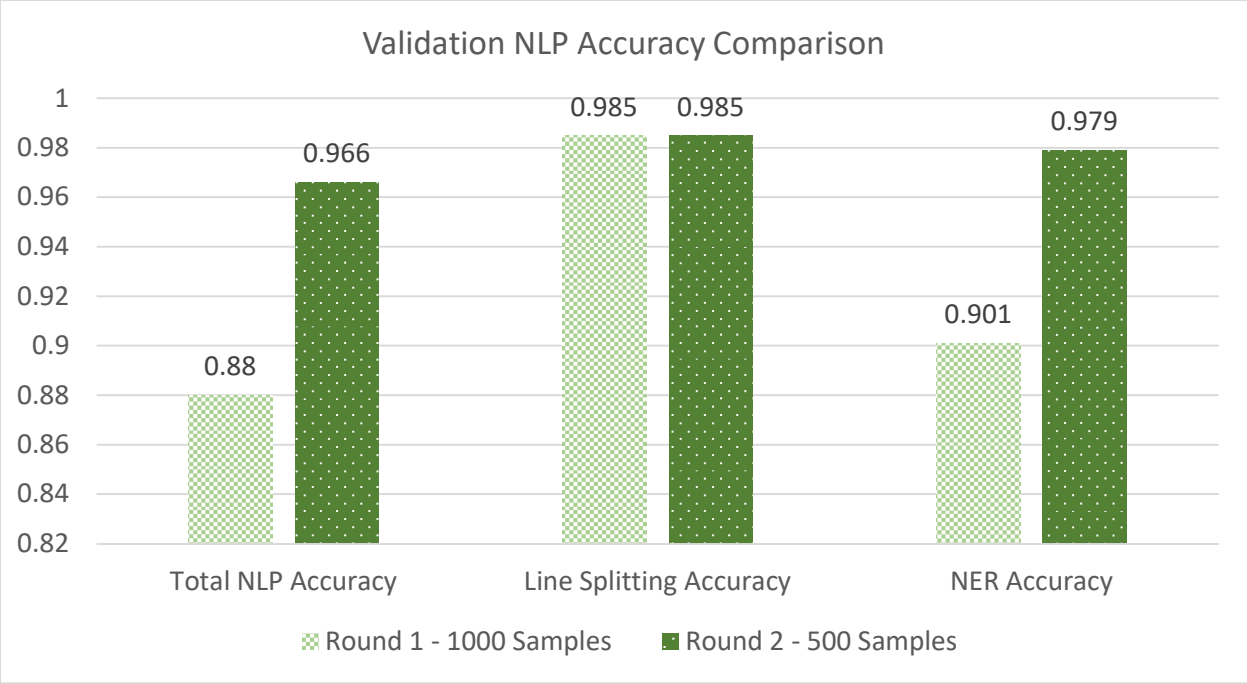


Fig. 7: Overall Validation Results

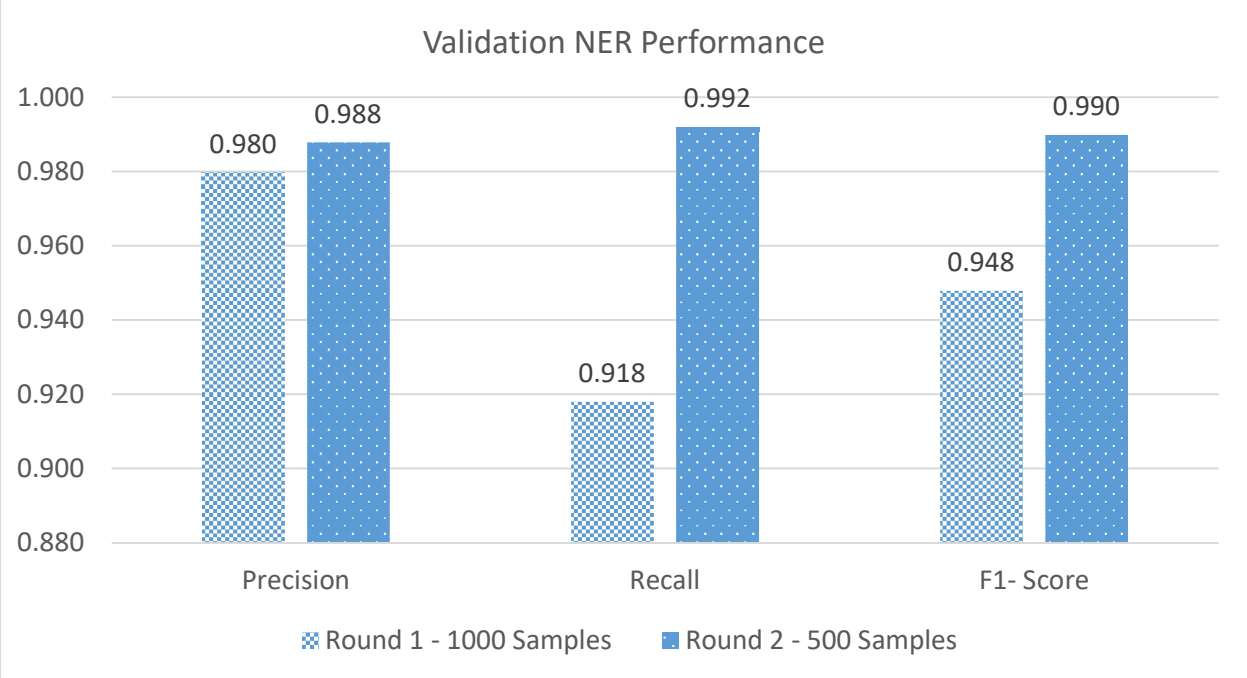


Fig. 8: Validation NER Performance

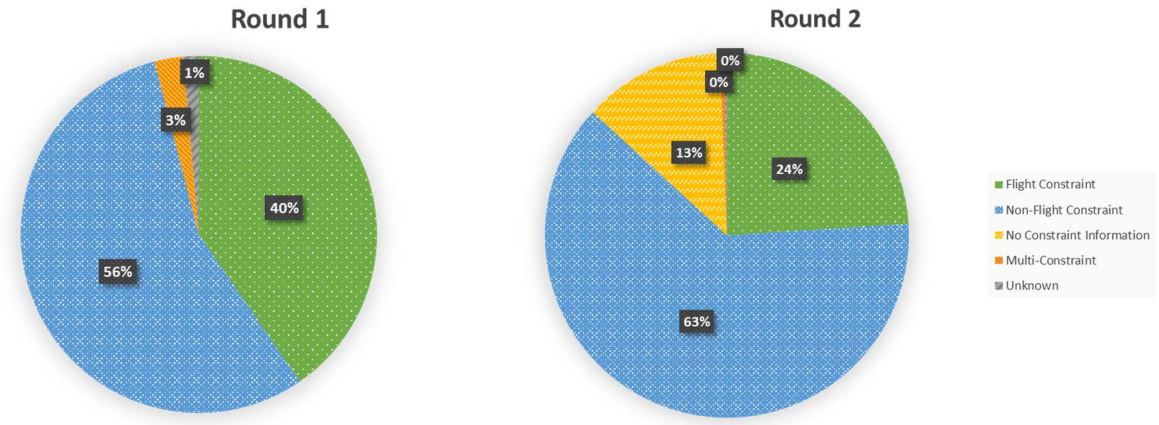


Fig. 9: Validation Constraint Types

Fig. 9 shows the distribution of constraint types per data sample in both rounds of validation. To reiterate, constraint code 7, or 'no constraint information,' was not considered in round one which is why it does not appear in the round one results. It is seen that there are roughly 24-40% flight constraints contained within the given data. That means there are roughly between 5,738 and 9,564 relevant flight constraints for potential extraction. There are also in the order of 56-63% of non-flight constraints. These constraints may be targeted in future iterations of the research for identification and extraction. While 3% of samples contain multi-constraints in round one, very few (< 1%) of samples contain multi-constraints in the second round. This issue arises in round one mostly due to line-splitting errors, where multiple lines were mistakenly considered as one, in turn causing multiple constraints to be included in a single data sample. Now that we understand the data distribution better and having validated the NER method, the next step is to identify specific flight constraints given the extracted entity information.

VI. Constraint Identification, Representation and Extraction

Recall that constraints in the context of LOAs can be defined as required procedures, operations or restrictions posed on pilots, operators, other persons, facilities, and stakeholders involved in a document. Starting from the tagged LOA data, the constraints of interest now need to be identified and extracted. Standard approaches to this problem involve techniques such as manual searching and unsupervised pattern finding. Also recall the challenges of working with natural language data. There are many ways to describe a flight constraint, and the goal is to generalize them into defined classes.

Another big challenge in creating these templates is determining the specificity of each individual template. If a template is too specific, it will only cover a very small percentage of the total data. If a template is too broad, it may not capture meaningful constraint information. Note that constraint identification is currently an ongoing process, so as more patterns around found, the specificity may change to cover more or less of the data.

A. Manual Constraint Identification/Extraction

Manual identification involves filtering and traversing the tagged LOA line level dataset in search for common constraint patterns. This dataset has features for the document name, document text, line key, tagged text, tokenized list of words, and list of entities. Therefore, there is easy querying of entity names or specific words looking to be found. For example, one could filter by the entity AERODROME and word 'arrival' to find lines regarding airport arrivals. These can then be filtered further depending on any patterns found. With prior knowledge and subject matter expertise, this process is much quicker and generalized patterns can be found.

B. Unsupervised Pattern Finding Using Clustering

The objective of clustering is to find collections of samples in unlabeled data that are similar to other samples within the cluster, and dissimilar to samples in other clusters [13]. In the problem we are interested in, clusters should represent specific categories of LOA constraints. Note that since the objective is to capture flight constraints, each

cluster must be analyzed for relevant flight constraints. Since the constraint clusters are not initially known, a density-based clustering method (e.g., HDBSCAN [14] algorithm) is used to identify the diverse set of potential patterns [13].

The workflow for the clustering process is as follows:

1. Replace entity-specific words with generalized named entity ("EWR" is replaced by "AERODROME").
2. Create token embeddings from last hidden-layer output of baseline and fine-tuned transformer model RoBERTa.
3. Use Uniform Manifold Approximation & Projection (UMAP) [15] algorithm to reduce dimensionality of data. This is done because the generated RoBERTa embeddings contain 768 dimensions.
4. Use HDBSCAN algorithm to cluster embeddings.
5. Perform another round of UMAP dimensionality reduction to two dimensions for visualization.
6. Repeat the process to generate sub-clusters if applicable.

Within this process, there are a few hyperparameters to tune and adjust to achieve the best performance. One of these hyperparameters is the initial UMAP dimensionality reduction. This typically ranges between 10 and 50 for optimal clusters. This is chosen to be not too small, as to not lose too much information of the data, but not too big because HDBSCAN works poorly in very high dimensional data [15]. Other UMAP parameters include the number of neighbors, the minimum distance separation between close embedding points, and the number of training epochs [15]. Some parameters for HDBSCAN are the minimum cluster size, minimum sample size and cluster selection epsilon [14].

C. Extracted Template Example

The following is an example of an *extracted* and *defined* constraint pattern template. Words displayed in brackets represent named entities that would be placed in the template, extracted from the original text. Also note that the template does not follow the original text word-for-word.

"Arrivals landing at [AERODROME] must use [FIX] STAR[11]"

Here are two data samples that fall into this template from our dataset[12]:

1. "… Arrivals to SFO, requesting at or above 10,000 feet MSL, must be routed via CEDES STAR."
2. "Arrivals to ATL departing RIC or airports north of RIC, requesting at or above 11,000 feet MSL must be routed via the appropriate STAR."

Given these examples, not all entities are captured within the template to make it more general. In example 1, arrivals are routed via the STAR, but only aircraft equipped with specific equipment. In example 2, only arrivals departing specific aerodromes and requesting a specific altitude are routed via the STAR. These additional restrictions can be thought of as *optional fields* to the current template and will be defined in later iterations of this research.

To date, through manual pattern finding and clustering, we have classified close to 500 data samples into templates. Since there is estimated to be around 5000-9000 samples that contain flight constraints within our dataset (concluded from the validation effort), these patterns account for about 5.5% to 10% of the data. This is ongoing work, and we plan to significantly increase the amount of data represented by pattern templates.

D. Future Work

Although we have made significant progress in understanding the distribution of constraints found within the data, there is still more work to be done. Each of the current patterns have covered roughly 70-100 data samples. Since these are the first patterns to be found, it is likely that the undiscovered patterns will represent less and less data samples. Through our initial clustering work, we see that the clusters typically contain a small number of samples compared to the size of the entire dataset (the largest clusters are estimated to cover less than 1% of the data), so a lot more effort is needed to define each cluster. We plan to continue the clustering process until we describe no less than 50% of the flight constraints in the data using pattern templates.

[11] STAR: Standard Arrival Routes

[12] These are tailored examples which are representative of patterns found within the LOA documents, but are not effective constraints pertaining to the facilities involved.

Along with pattern templates, we plan to investigate NLP tools like *entity linking* to generalize relationships between the named entities. Finally, the overarching goal will be adapting these constraints into an operational format like AIXM. This may require modifications or additions to AIXM data structures and allowed data formats. This will be a part of our ongoing collaboration with the AIXM development community and the FAA.

VII. Conclusions

In this work, we have applied traditional natural language processing methods, proven on generic document text, to extract flight constraint information from Letters of Agreement, which are legacy air traffic management documents. We used AWS's Textract to parse the pdf source documents with acceptable accuracy. An appropriate *data unit* selection was made such that it contains no more than one flight constraint. After cleaning up and proper preprocessing of the dataset, we used gazetteers and syntactic-lexical patterns to extract named entities pertaining to flight constraints with a validated F1-score of 99.0%. Most importantly, a preliminary set of constraints have been *identified* (the last step in the process towards representation in XM format) as generic entity patterns with the aid of deep-learning transformer models like RoBERTa. The analysis of the LOA dataset shows that the text contains many pre-defined and well-known entities. Therefore, traditional NER methods worked and resulted in remarkable accuracy better than what has been achieved on open domain data [10].

The work presented in this paper documents our initial findings on the challenging task of digitizing flight constraints embedded in legacy aviation documents. We plan to continue the work and additionally also transfer the methods developed to other heritage air traffic management documents like Standard Operating Procedures (SOPs).

Acknowledgments

We thank the FAA for providing the LOA documents, supporting the research with subject-matter expertise, validating our results and engaging in quality discussions throughout the process.

References

[1] E. Porosnicu, B. Murphy and K. Wilson, "Aeronautical Information Exchange Model," Eurocontrol, [Online]. Available: https://www.aixm.aero/.

[2] "7210.3CC Section 3. Letters of Agreement(LOA) pp. 143-150," FAA, 17 June 2021. [Online]. Available: https://www.faa.gov/documentLibrary/media/Order/7210.3BB_FAC_Bsc_w_Chg_1_2_3_dtd_12-31-20_For_Post.pdf.

[3] A. Doan, R. Ramakrishnan and S. Vaithyanathan, "Managing information extraction: state of the art and research directions.," in *2006 ACM SIGMOD international conference on Management of data*, 2006.

[4] "NLP Fact Sheet," Mosaic ATM, 2021. [Online]. Available: https://mosaicatm.com/wp-content/uploads/2022/02/matm-nlp-fact-sheet.pdf.

[5] "This is NextGen," Federal Aviation Administration, 05 January 2022. [Online]. Available: https://www.faa.gov/nextgen/this_is_nextgen/.

[6] Y. Zhao, H. Liu and Z. Chen, "Named Entity Recognition for Chinese Aviation Security Incident Based on BiLSTM and CRF," in *Asia Conference on Computers and Communications*, 2021.

[7] R. L. Rose, T. G. Puranik and D. N. Mavris, "Natural Language Processing Based Method for Clustering and Analysis of Aviation Safety Narratives," *MDPI Aerospace,* 2020.

[8] C. Posse, B. Matzke, C. Anderson, A. Brothers, M. Matzke and T. Ferryman, "Extracting Information from Narratives: An Application to Aviation Safety Reports," in *IEEE Aerospace Conference*, 2005.

[9] J. Li, A. Sun, J. Han and C. Li, "A Survey on Deep Learning for Named Entity Recognition," *IEEE Transactions on Knowledge and Data Engineering,* vol. 34, no. 1, pp. 50-70, January 2020.

[10] S. Sekine and C. Nobata, "Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, Lisbon, Portugal, 2004.

[11] P. Bhatia, B. Celikkaya, M. Khalilia and S. Senthivel, "Comprehend Medical: a Named Entity Recognition and Relationship Extraction Web Service," in *IEEE International Conference on Machine Learning and Applications*, 2019.

[12] R. M. Keller, "Building a Knowledge Graph for the Air Traffic Management Community," in *Companion Proceedings of the 2019 World Wide Web Conference*, 2019.

[13] T. S. Madhulatha, *An Overview On Clustering Methods,* vol. 2, IOSR Journal of Engineering, 2012, pp. 719-725.

[14] R. J. G. B. Campello, D. Moulavi and J. Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," in *Advances in Knowledge Discovery and Data Mining*, 2013.

[15] L. McInnes, J. Healy and J. Melville, *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction,* 2018.

[16] M. Bates, *Models of Natrual Language Understanding,* vol. 92, National Academy of Science, 1995, pp. 9977-9982.

[17] G. Zaman, H. Mahdin, K. Hussain and A. Ur-Rahman, *Information Extraction From Semi and Unstructured Data Sources: A Systematic Literature Review,* ICIC , 2020.

[18] X. Wang, H. Ding and L. Zeyuan, *Information Extraction of Air-Traffic Control Instructions via Pre-trained Models,* Artificial Intelligence in China, 2022.

[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and S. Veselin, *RoBERTa: A Robustly Optimized BERT Pretraining Approach,* arXiv, 2019.