Natural Language Processing Analysis of Notices To Airmen for Air Traffic Management Optimization

Stephen S. B. Clarke[*] and Patrick Maynard[†]
Sacred Heart University, Fairfield, CT, 06825
Harvard University, Cambridge, MA, 01238

Jacqueline A. Almache[‡]
North Carolina State University, Raleigh, NC, 27695

Satvik G. Kumar[§]
Georgia Institute of Technology, Atlanta, GA, 30332

Swetha Rajkumar[¶]
Lynbrook High School, San Jose, CA, 95129

Alexandra C. Kemp[‖]
Purdue University, West Lafayette, IN, 47906

Raj Pai[**]
National Aeronautics and Space Administration, Mountain View, CA, 94035

With new emerging technologies in the field of NLP, we explore their applications to digitize and analyze heritage Air Traffic Management (ATM) documents for planning and optimizing airspace operations. Specifically, this research focuses on harvesting semi-structured or unstructured information contained in Notices to Airmen (NOTAMs). Using NLP and other advanced data analytics, we will construct a data-driven framework which facilitates finding language patterns and the use of pretrained language models for classification and extraction of useful airspace constraints and restrictions. These may lead to tools that assist airspace users in understanding the constraints more efficiently, contributing to better route planning and safer execution. This paper explores three workflows entailing different NLP tasks. First, unsupervised techniques like word embedding and topic modeling are used for pattern finding and document classification. Second, a dataset is created by extracting information from the semi-structured NOTAM format as metadata for categorizing, visualizing, and extracting key entities driving NOTAM content. Third, modern pre-built deep learning based transformer models such as BERT, RoBERTa, and XLNet are evaluated on the question answering task, an even more robust approach to information extraction, as well as their respective fine-tuning tasks. In this work we include various performance metrics for the trained models to evaluate both accuracy and precision and we show that the models can be generalized for their respective tasks. The research work developed shows promise in uncovering trends in digital NOTAMs in the NAS and also offers a new framework for digitizing and inferring insights from free-form legacy NOTAMs, that are yet to be digitized.

[*]Graduate Student, School of Computer Science and Engineering, Sacred Heart University, Fairfield, CT 06825, AIAA Student Member

[†]Undergraduate Student, Harvard Extension School, Harvard University, Cambridge, MA 01238, AIAA Student Member.

[‡]Graduate Student, Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, 27695, AIAA Student Member

[§]Undergraduate Student, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, AIAA Student Member

[¶]High School Student, Lynbrook High School, San Jose, CA, 95129

[‖]PhD Student, Purdue Polytechnic Institute, School of Aviation and Transportation Technology, Purdue University, West Lafayette, IN, 47906, AIAA Student Member

[**]Research Fellow, NASA Aeronautics Research Institute, Moffett Blvd, Mountain View, CA 94035, and AIAA Member

I. Introduction

Notice to airmen (NOTAMs) contain vital information for all aviation stakeholders and are critical for personnel involved with flight operations. A NOTAM states the abnormal status of some component found in the National Airspace System (NAS). A NOTAM is typically related to facility, weather condition, service procedure, hazard conditions that impact the normal status of the NAS. Originally introduced in 1947 [1], NOTAMs have evolved into a unique language that uses special constructs, acronyms, and English abbreviations to make communication more efficient.

Recently in 2011, the digital NOTAM was first adopted by the Federal Aviation Administration (FAA) following the Aeronautical Information Exchange Model (AIXM) 5.1 specification [2]. This model describes NOTAMs through 'scenarios' which are business rules that attempts to describe all situations that can arise in the NAS. These rules are all documented in the Digital NOTAM Specification [2], and future global standards are also evolving[*]. The digital NOTAM has paved the way to research including systems to appropriately filter and query relevant NOTAMs [3]. Although the AIXM model digitally captures most NOTAMs, roughly 30% are not yet digitized. The majority of this problem has occurred due to insufficient infrastructure at smaller airports, as well as time and resource constraints for certain types of heritage NOTAMs that have yet to be modeled by AIXM. While digital NOTAMs have made it easier to check for NOTAMs prior to a flight, the increasing volume of NOTAMs and advances in deep learning and natural language processing techniques offer new opportunities to understand and optimize NOTAMs to the benefit of pilots, air traffic managers, and researchers.

II. Background

Natural language processing(NLP) includes computerized techniques that are used to analyze and represent language [4, 5]. Recently, NLP techniques have been utilizing advances in the field of machine learning, deep learning, and artificial intelligence for various tasks [5]. NLP techniques include but are not limited to word embeddings to transform words into vectors, which are usually the first data processing layer [5], to topic modeling [6], named entity recognition (NER) which is "used to extract entity objects with certain meanings from text data" [7], language modeling such as bidirectional encoder representations from transformers (BERT) [8], or question answering [9].

With the modernization and growth of air traffic in the past decade [10], air traffic control (ATC) is facing a challenge to modernize [11]. This is evident in the United States with initiatives such as NextGen [11]. Stroup et al. have looked into ways artificial intelligence can be applied to the NAS [12] as well as the rationale for the use of AI in the NAS [13]. They note natural language processing as a technique that could be used. As traditional documents such as NOTAMs, LOAs, and SOPs are still widely used today, it is worth examining how advances in AI and NLP can be used to help with the modernization of the NAS.

Existing research in the aviation domain mostly focuses on using NLP techniques to help pilots with aircraft maintenance issues or for aviation safety analysis. Paul et al. [14] investigated NLP techniques that are used in Civil Aviation. They focused on applications such as IBM Watson's question and answering tool to help pilots, flight attendants, technical staff, and customer service agents with maintenance challenges and Boeing's NLP system (BLUE) to generate a semantic representation for texts such as technical and maintenance manuals [15]. Paul et al. [14] also mention data sources used for NLP such as the European Coordination Centre for Accident and Incident Reporting Systems (ECCAIRS) or the NASA Aviation Safety Report (ASRS) to collect and standardize reports of incidents in the European Union and United States respectively.

There are a multitude of studies that focus on the ASRS data set that use different NLP techniques. Rose et al. [16] presented a methodology to analyze aviation safety narratives using a word embedding technique called TF-IDF and clustering with k-means. By using this method, they were able to identify relationships between narratives which could also lead to more in depth analysis of certain groups. Similarly, Kuhn [6] explored the use of a topic modeling technique called Latent Dirichlet Allocation (LDA) in ASRS data to identify topics which can expose known and unknown issues and can pave the way to improve safety. Kierszbaum and Lapasset [17] take a slightly different approach with the ASRS data. They examine using BERT for a question and answering task. They used thirty randomly selected reports to answer the question "When did the incident happen" and they received 22 "good answers".

A lot of other high level work has also been done that focus on aviation safety reports. Tanguy et al. [18] examine different tools to analyze aviation safety documents through the use of support vector machines (SVM) and LDA topic modeling. Srinivasan et al. [19] propose a method to classify reports into binary classes using a combination of word embedding and the Long Short-term Memory (LSTM) algorithm. Madeira et al. [20] developed an NLP pipeline

[*]https://ext.eurocontrol.int/aixm_confluence/display/DNOTAM/Digital+NOTAM+Specification

to predict human factors in aviation safety incidents. Their pipeline included pre-processing, feature extraction with TF-IDF, Word2Vec, and Doc2Vec, and data modelling using semi-supervised Label Spreading(Ls) and supervised SVM.

Other NLP techniques such as named entity recognition (NER) have loosely been used in the aviation domain. Xing et al. [21] use NER to locate aviation customer service issues by examining civil aviation reviews. Finally, Bravin et al. [22] did do some work related to NOTAMs. They examined the use of a Seq2Seq model to automatically "smartify" NOTAMs in Europe and found promising results. It is important to note however, that NOTAMs published in the United States are coded in a separate domestic format which differs from the *International Civil Aviation Organization*, or ICAO format [23].

While there exists a multitude of research on the application of different NLP techniques to the aviation domain, much of this research thus far has been focused on the analysis of safety events with some other applications in aviation flight manuals or maintenance. Very limited research exists in literature regarding the use of NLP on heritage air traffic management documents. This paper incorporates many of the methods discussed in prior work to present a pipeline for the use of NLP on heritage air traffic management documents and to present several key findings when using these techniques on NOTAMs.

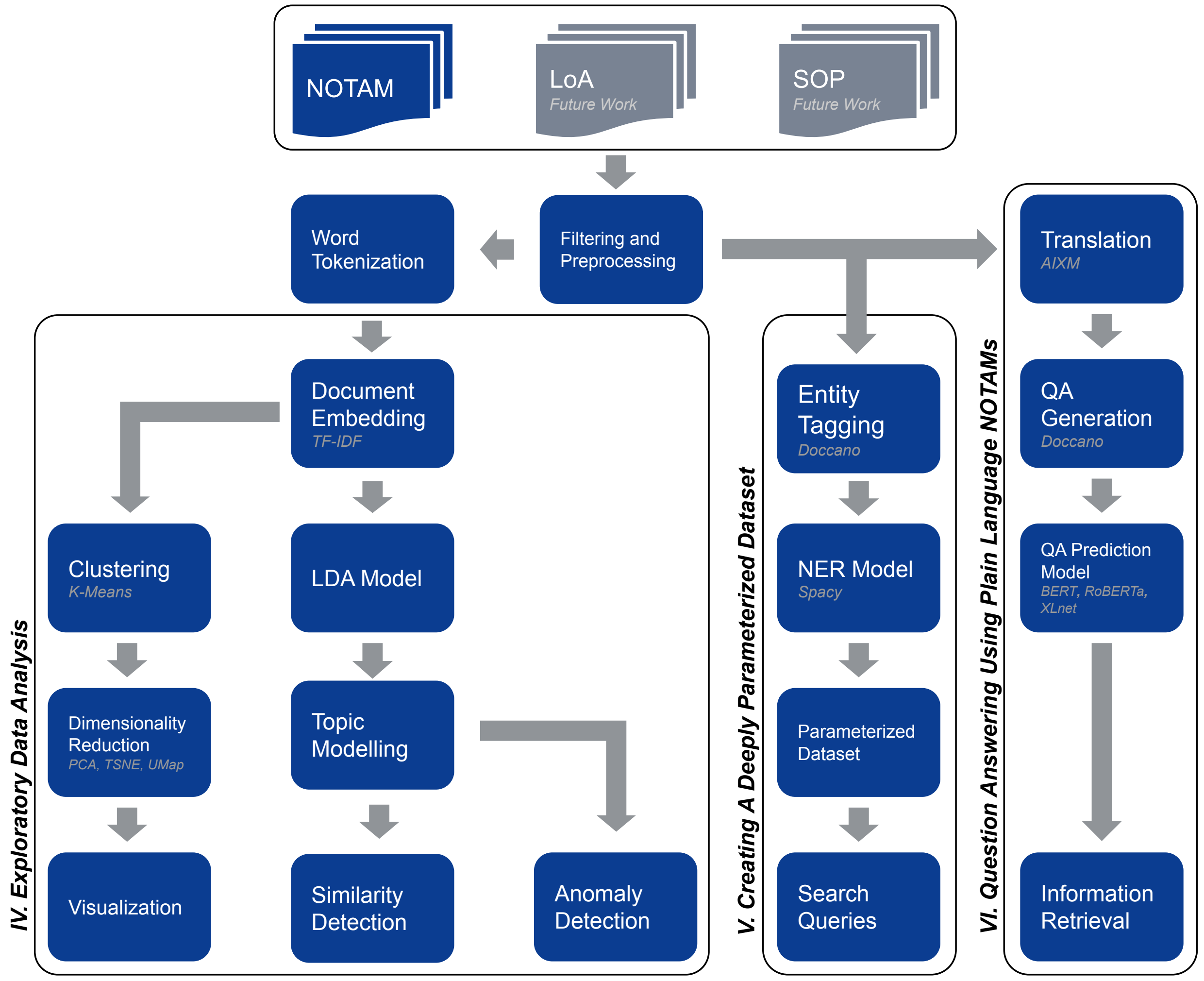

Fig. 1 Overview of Methodologies

The end goal of this research was to gain a deeper understanding about the current state of NOTAMs, and assess the capability of modern deep learning models for information retrieval from hand-written NOTAMs. Figure 1 gives an overview of the methodologies used to reach these goals. First, the raw data and preliminary pre-processing of NOTAM data sets will be discussed (Section III). Second, multiple exploratory data analysis tasks were performed to gauge pattern finding and understanding within and between different NOTAMs (Section IV). Clustering techniques were used to find these patterns, while topic modeling techniques like Latent Dirichlet Allocation, or LDA, were used to understand differences and relationships between documents. Third, given patterns and an idea of what can be extracted from NOTAMs, we moved onto our next objective, entity tagging (Section V). This tagging was used as both a tool for extracting specific entities, as well as defining a relationship structure within each individual NOTAM. Lastly, we assessed large deep learning models' ability on comprehension and information retrieval of these domain-specific documents (Section VI). Future efforts may include adopting the methodology used for NOTAMs for additional air traffic management (ATM) documents, digitizing hand-written NOTAMs, and creating hybrid systems that utilize both deep learning, and more rigid information exchange models.

III. Data and Preprocessing

A. NOTAM Background and Structure

Notices to Airmen, or NOTAMs, are short, highly contracted messages that describe real-time abnormal statuses, conditions, and changes within the NAS[24]. Listed below is an example of a NOTAM posted at Newark Liberty International Airport (EWR) pulled from the public FAA NOTAM Search website.[†] Typically, the audience for these documents are pilots, air traffic controllers, and other operational personnel involved in flight operations [2]. One of the biggest users of NOTAMs are pilots who brief on all active NOTAMs along a flight path before departing.

> !EWR 01/020 EWR TWY EE HLDG PSN MARKINGS FOR ILS BTN RWY 04R/22L AND TWY M FADED 2101041504-2106302300

This format, although confusing at first, is easily readable by someone familiar with NOTAMs. This one in particular is regarding EWR, or Newark International Airport, and states that the holding position markings on taxiway EE and M between runway 04R and 22L are faded. There is also specific information such as the start and end times which appear as the last two tokens in every NOTAM. They follow the format of two-digit year, month, day, hour and minute.

Due to the digital NOTAM structure, one can generate plain English versions for those less familiar with the NOTAM format. We use this plain English format for inputs into pretrained deep learning models as discussed later in section VI.

> Issuing Airport: (EWR) Newark Liberty Intl
> NOTAM Number: 01/020
> Effective Time Frame
> Beginning: Monday, January 4, 2021 1504 (UTC)
> Ending: Wednesday, June 30, 2021 2300 (UTC)
> Affected Areas
> Taxiway: EE (between RWY 04R/22L and TWY M)
> Marking Type: Holding position markings for ILS
> Status: Faded

Figure 2 gives a deeper understanding of the general structure found within the body of domestic NOTAMs [25]. Closed boxes represent required features, whereas dotted boxes are optional [25]. Overall, these features are used to describe how a specific attribute within the NAS is affected, giving descriptions and exact details of time and location. Recalling the example above, the **keyword** is TWY, the **attribute** is EE HLDG PSN MARKINGS (referring to holding position markings on the taxiway named EE), the **condition** is FADED, and the **start of activity/end of validity** times are shown by the last two numbers: 2101041504 and 2106302300.

This structure is explored further when creating a parameterized dataset in Section V, where each box in Fig. 2 can represent a unique feature of a NOTAM.

B. Our Data

In order to perform machine learning on NOTAM data, a collection of NOTAMs is needed. The dataset used for the studies in this paper is provided by the FAA. Specifically, this data includes 3.73 million NOTAMs active between

[†]`https://notams.aim.faa.gov/notamSearch`

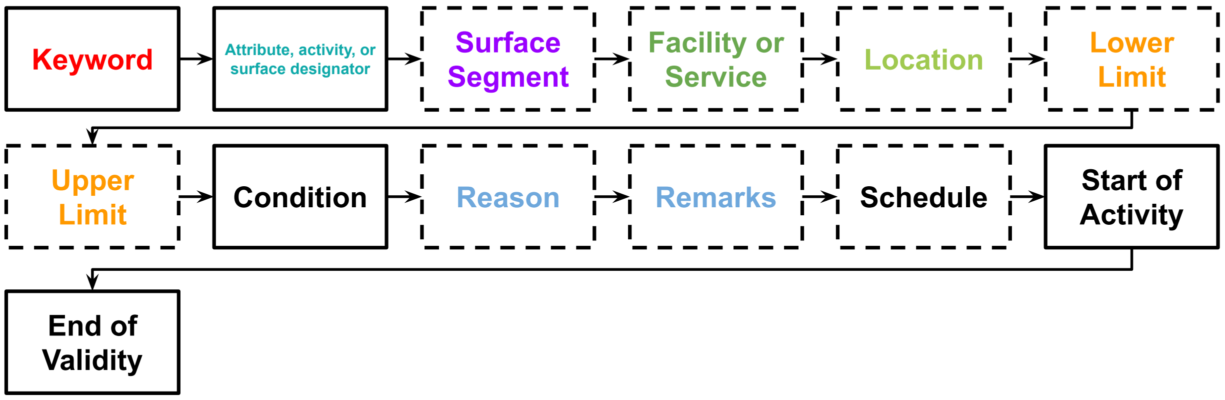

Fig. 2 General NOTAM structure [25].

March 2019 and March 2020. These NOTAMs include roughly 30.34% flight data center [‡] and and 69.66% domestic NOTAMs. This dataset also includes 1.04 million digital NOTAMs that are converted into ICAO and Plain Language formats. The Plain Language format will be used in Section VI for deep learning tasks.

C. Preprocessing

Before performing any tasks on the data, a few preprocessing steps were taken to ensure both consistency between data and best performance when creating models. As different NLP tasks require different forms of preprocessing, additional optional steps taken are mentioned.

1) Completely remove non-conforming NOTAMs (about 60,000 of 3.7 million or 1.6%) which:
    1) Do not contain valid start/end times. Valid timestamps follow the format YYMMDDHHmm[§].
    2) Do not contain a valid NOTAM number. Valid NOTAM numbers follow the format MM/####.
    3) Do not contain a valid keyword. Valid keywords are RWY, TWY, APRON, AD, OBST, NAV, COM, SVC and AIRSPACE.
2) Remove all non-alphanumeric characters.
3) Separate all tokens by spaces.
4) **(Optional)** Convert all characters to lowercase.
5) **(Optional)** Replace specific numeric values with special tokens such as *time* or *notam_num*, or remove numbers all-together. Wallace [26] shows that certain NLP models have difficulty understanding large numbers, especially outside of their training sets, which NOTAMs are polluted with. Some of the models in this paper also experienced metric gains when removing numeric values.

IV. Exploratory Data Analysis

In this section, the raw NOTAM data are explored for inter-document patterns, similarities and anomalies. The methodology includes converting each NOTAM into an embedding, or high-dimensional vector, to represent it numerically. There are algorithms to convert words into vectors, and in this paper we primarily discuss a simpler statistical technique called *Term Frequency-Inverse Document Frequency*. After embedding the data, unsupervised clustering algorithms such as k-means or *Agglomerative Hierarchical Clustering* are used to create groupings of similar NOTAMs. These vectorized documents are then passed into a dimensionality reduction algorithm for visualization. The dimensionality reduction algorithms explored are *Principal Component Analysis* (PCA), *t-Distributed Stochastic Neighbor Embedding* (t-SNE), and *Uniform Manifold Approximation and Projection* (UMAP). Topic modeling along with similarity and anomaly detection operations are also performed on the TF-IDF embedded data using *Latent Dirichlet Allocation* (LDA).

[‡]All flight data center NOTAMs start with FDC. They typically include changes to IFR flight procedures or temporary flight restrictions [25].
[§]YYMMDDHHmm: Two digit year, month, day, hour, minute

A. Word Embedding

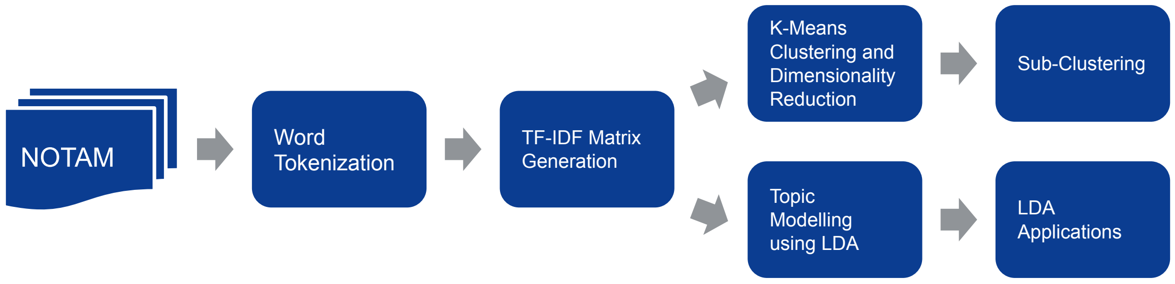

Fig. 3 Methodology that was followed to gain insight from TF-IDF word embedding.

TF-IDF is a statistical technique that can determine the relevance and significance of certain words in a particular document [27]. Figure 3 summarizes the methodology that was followed to gain insight from TF-IDF word embedding and will be explained in detail throughout this section. First, a matrix is generated in which each row refers to a NOTAM, and each column refers to a different word. All unique words occurring in the entire corpus will have a column. Counts of each word in a particular NOTAM are placed in the appropriate row/column. The matrix generated is commonly referred to as a Bag of Words (BoW) matrix. An example is shown in Table 1.

Table 1 Example Bag of Words Matrix

	$Word_1$	$Word_2$	...	$Word_i$
$NOTAM_1$	$Count_{1,1}$	$Count_{2,1}$	...	$Count_{i,1}$
$NOTAM_2$	$Count_{1,2}$	$Count_{2,2}$	...	$Count_{i,2}$
...	...	...	...	...
$NOTAM_j$	$Count_{1,j}$	$Count_{2,j}$	...	$Count_{i,j}$

To generate the TF-IDF matrix, Equation 1 was applied to every cell *i,j* where *i* represents each word and *j* represents each NOTAM:

$$tfidf_{i,j} = tf_{i,j} \cdot \log\left(\left(\frac{N+1}{1+df_i}\right)+1\right) \tag{1}$$

where $tfidf_{i,j}$ is the TF-IDF score of the ith word in document *j*, $tf_{i,j}$ is the term frequency or the number of occurrences word *i* in document *j*, N is the number of NOTAMs, and df_i is the number of documents word *i* appears in. The TF-IDF score vector for each NOTAM *j* is then normalized using the Euclidean norm. The TF-IDF matrices for the NOTAM data-set were generated through the *TfidfVectorizer* function in scikit-learn's feature extraction module [¶]. The default parameters were used. This function uses Equation 1, a slightly modified version of the standard TF-IDF equation found in most textbooks and other works [28]. The scikit-learn feature extraction module documentation[¶] can be consulted for further explanation of the equation. All NOTAMs were pre-processed so that every number was replaced with the # symbol using Python's regular expression operations function [‖]. An example of a TF-IDF matrix is shown in Table 2.

The generated TF-IDF matrix contained 5,000 randomly selected NOTAMs from the initial 3.73 million NOTAMs. NOTAMs were down-sampled to 5,000 as running these algorithms on all NOTAMs would be computationally expensive.

B. K-Means Clustering

K-means is a clustering algorithm that is designed to split data into exactly k number of clusters [29]. The algorithm places points in clusters to reduce the squared Euclidean distance between points in the same cluster. The Euclidean

[¶]`https://scikit-learn.org/stable/modules/feature_extraction.html`
[‖]`https://docs.python.org/3/library/re.html`

Table 2 Example TF-IDF Matrix

	$Word_1$	$Word_2$	...	$Word_i$
$NOTAM_1$	$tfidf_{1,1}$	$tfidf_{2,1}$	...	$tfidf_{i,1}$
$NOTAM_2$	$tfidf_{1,2}$	$tfidf_{2,2}$	...	$tfidf_{i,2}$
...	...	...	...	...
$NOTAM_j$	$tfidf_{1,j}$	$tfidf_{2,j}$	...	$tfidf_{i,j}$

distance is the root of square difference between two coordinates [30]. In this case, NOTAMs that share similar words and length will be "closer". The use of the k-means algorithm on a TF-IDF matrix for aviation related safety narratives has had prior success [16].

The k-means clustering algorithm was run on the generated TF-IDF matrix using scikit-learn's clustering module [**]. Values of k (number of clusters) from two to ten were chosen and run. The clusters and TF-IDF matrix were visualized using PCA, t-SNE, and UMAP to reduce the dimensionality of the TF-IDF matrix to two for visualization on a two-dimensional plane. Reducing dimensionality to three-dimensions was also tried, but did not provide significantly more insight than two-dimensions. PCA [31], t-SNE [32], and UMAP [33] are commonly used methods to reduce dimensionality and reveal structure in data. PCA dimensionality reduction was implemented using scikit-learn's decomposition module [††]. Scikit-learn's manifold module was used to implement t-SNE [‡‡]. The UMAP-Learn module [§§] was used to implement UMAP dimensionality reduction. Both t-SNE and UMAP have hyperparameters that can be tuned. For t-SNE, the *perplexity* hyperparameter was set to 300, *learning_rate* was set to 50, and *random_state* was set to 20. All other hyperparameters were left at their default values. For UMAP, *n_neighbors* was set to 45 and the *random_state* was set to 20. All other hyperparameters were left at their default values. An arbitrary value was chosen for the random state for reproducibility. Multiple values each of the hyperparameters were chosen and tested to find values that produced meaningful visualizations.

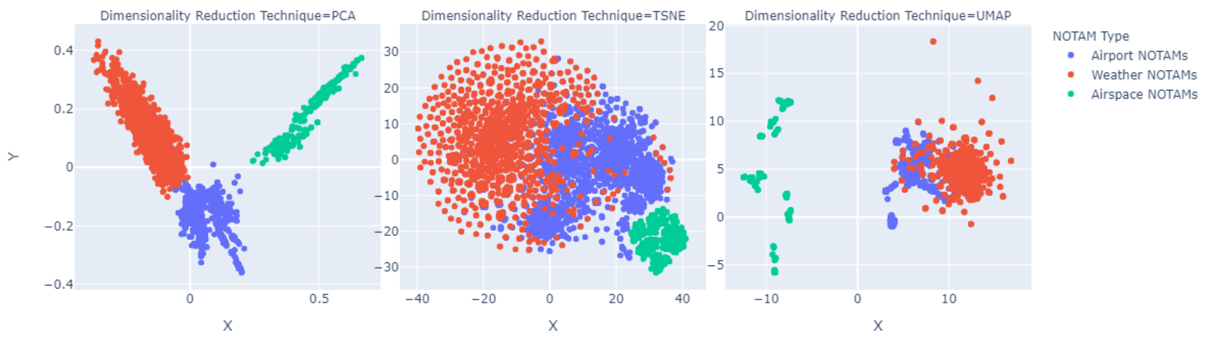

Fig. 4 Visualization of TF-IDF Matrix on 5,000 NOTAMs following dimensionality reduction. Clusters are found using the k-Means clustering algorithm with a k-Value of 3.

Figure 4 shows results for k-means clustering with a k-value of 3 after PCA, t-SNE, and UMAP dimensionality reduction. Table 3 summarizes the top 20 keywords found in each cluster based on the TF-IDF matrix. Most success was found breaking the NOTAMs into three clusters as the NOTAMs were clustered into three distinct categories: Airport NOTAMs, Weather NOTAMs, and Airspace NOTAMs. These categories were determined based on the top keywords from each cluster and by examining NOTAMs that were in these various clusters. These three categories encompass the reasons a majority of NOTAMs are issued. There were 2,583 NOTAMs grouped into the Weather category, 1,842 NOTAMs grouped into the Airport category, and 575 NOTAMs grouped into the airspace category. Choosing a value

[**]`https://scikit-learn.org/stable/modules/clustering.html`
[††]`https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`
[‡‡]`https://scikit-learn.org/stable/modules/manifold.html`
[§§]`https://umap-learn.readthedocs.io/en/latest/`

Table 3 TF-IDF K-Means Clustering Top 20 Words

Category	Top 20 Words
Airport NOTAMs	ils, btn, tower, sfc, us, obst, nav, agl, wip, lgt, ad, ap, nm, ft, out, service, of, rwy, twy, clsd
Weather NOTAMs	medium, ft, ba, over, wid, and, twy, apron, ice, all, compacted, dry, rwy, wet, pct, in, sn, at, ficon, obs
Airspace NOTAMs	zkc, zlc, zab, sfcfl, ar, sfc, zse, not, to, suae, up, but, including, moa, suac, suaw, ft, fl, airspace, act

of k greater than 3 did not produce more meaningful clusters while a k value of 2 combined the Weather and Airport NOTAM clusters into the same cluster. The Airport NOTAM category includes but is not limited to NOTAMs that refer to the state of operations at various airports such as runway and taxiway closures, obstructions, navigation using the instrument landing system, etc. This is signified by the top keywords in this category such as “ils” which stands for instrument landing system, “obst” which stands for obstruction, “nav” which stands for navigation, “rwy” and “twy” which stand for runway and taxiway respectively, “clsd” which stands for closed, and the keywords “out,” “of,” and “service.”

The Weather NOTAM category includes but is not limited to NOTAMs that refer to weather conditions at airports. The most common weather NOTAMs in the data-set tend to refer to precipitation on runways and taxiways. This is signified by the top keywords in this category such as “dry,” “wet,” “ice,” “pct” which stands for percent, “wid” which stands for width, as well as the keywords such as “rwy,” “twy,” and “apron” which identify where these weather obstructions may be happening. As seen in both t-SNE and UMAP dimensionality reduction, there is overlap between Airport NOTAMs and Weather NOTAMs while PCA shows a clear split. However, even in PCA, Airport NOTAMs and Weather NOTAMs are close together. This split makes sense as Weather NOTAMs are applicable at airports. Additionally, many of the keywords associated with Weather NOTAMs such as runway and taxiway overlap with the keywords referred to in Airport NOTAMs.

The Airspace NOTAM category includes but is not limited to NOTAMs that refer to conditions in the airspace. Most of the NOTAMs in this category refer to restricted airspace or military operating areas. This is signified by top keywords such as “airspace,” “moa” which stands for military operating airspace, “suaw” and “suac” which stand for special use airspace west and special use airspace central respectively, “fl” which stands for flight-level, “ft” which stands for feet commonly referring to the specific altitudes the NOTAM applies to, and “up to but not including” which is a common phrase in these NOTAMs when referring to the altitudes that the NOTAM applies. Additionally, the top three keywords in this category refer to specific FAA airspaces/air route traffic control centers.

These clustering results indicate that TF-IDF scores of the words in each NOTAM are an effective way to split NOTAMs into different major categories with the use of k-means clustering.

1. Sub-Clustering

After initial clustering, the 5,000 random NOTAMs from the original data-set were separated by the cluster they were split into. A new matrix was generated for each cluster by taking the corresponding rows for every NOTAM in each cluster from the original TF-IDF matrix. This produced three different TF-IDF matrices for each of the different clusters. The k-means clustering algorithm discussed before was run on each of these different matrices again with different k values ranging from two to ten.

Table 4 Example Airspace NOTAMs in Each Cluster

Cluster	Example NOTAM
MOA NOTAMs	...AIRSPACE ADIRONDACK C MOA ACT 100FT AGL UP TO BUT NOT INCLUDING FL180...
	...AIRSPACE COLUMBUS 2 MOA ACT 8000FT UP TO BUT NOT INCLUDING FL180...
High Altitude Airspace NOTAMs	...AIRSPACE AR197L ACT FL200-FL220...
	...AIRSPACE W237A HIGH ACT FL230-FL350...
Other NOTAMs	...AIRSPACE R3008B ACT 100FT-10000FT...
	...AIRSPACE R6412C ACT SFC-9000FT...

Figure 5 shows results for k-means clustering with a k value of three for the airspace sub-cluster matrix. A k value of three provided the most meaningful results. Based on observing the NOTAMs in each cluster, it is noted that the red cluster mostly consists of NOTAMs that are issued for a military operating airspace (MOA). The green cluster contains

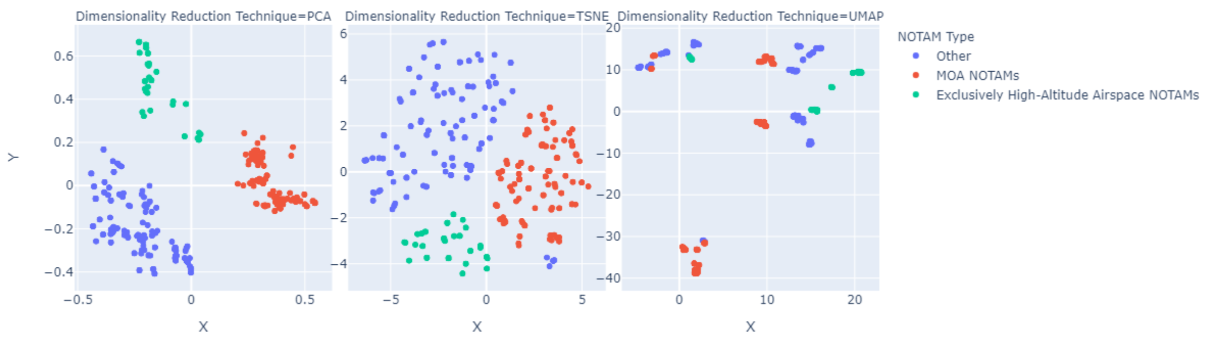

Fig. 5 Visualization of Airspace sub-cluster NOTAMs based on TF-IDF matrix following dimensionality reduction. Clusters are found using the k-Means clustering algorithm with a k-Value of 3.

special use airspace NOTAMs that tend to contain the phrase "FL###-FL###" where ### is a number that corresponds to the altitudes that these NOTAMs would apply at. FL means flight level and can be calculated by dividing the altitude in feet by 100. Flight level is generally used when discussing altitudes greater than 18,000 feet. Therefore, the green cluster consists of special use airspace NOTAMs that only apply at altitudes above 18,000 feet. The remaining special use airspace NOTAMs are represented by the blue cluster. These include NOTAMs that only apply at lower altitudes or NOTAMs that may begin at a lower altitude but continue onto a high altitude. Table 4 shows example NOTAMs in each cluster.

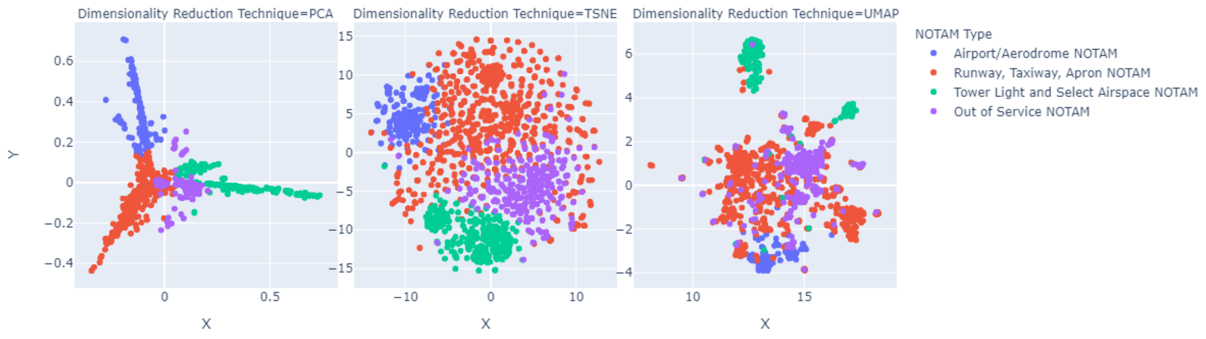

Fig. 6 Visualization of Airport sub-cluster NOTAMs based on TF-IDF matrix following dimensionality reduction. Clusters are found using the k-Means clustering algorithm with a k-Value of 4.

Table 5 Example Airport NOTAMs in Each Cluster

Cluster	Example NOTAM
Airport/Aerodrome NOTAM	...AD AP ALL SFC WIP GRASS CUTTING... ...AD AP ALL SFC WIP SN REMOVAL...
Runway, Taxiway, Apron NOTAM	...TWY H2 CLSD... ...RWY 091/27R CL MARKINGS OBSC... ...APRON PAPA PAD CLSD...
Tower, Light, and Select Airspace NOTAM	...OBST TOWER LGT (ASR 1002758) 393449.00N0862515.00W (10.1NM SW IND) 1249.0FT (419.0FT AGL) U/S... ...AIRSPACE UAS WI AN AREA DEFINED AS 0.5NM RADIUS OF 294641N950834.4W (7NM NW T41) SFC-340FT AGL...
Out of Service NOTAM	...NAV ILS RWY 28L OUT OF SERVICE... ...TWY B DIRECTION SIGN OUT OF SERVICE...

Figure 6 shows results for k-means clustering with a k value of four for the airport sub-cluster matrix. Results for a k value of four are shown here as this was able to find the most meaningful groupings in the airport sub-cluster. Table 5 shows example NOTAMs taken from each cluster and can be referenced to better understand the types of NOTAM that were grouped into each cluster. The blue cluster mostly consists of NOTAMs that consist of the keywords “AD AP” which stand for Airport and Aerodrome respectively. Therefore, this cluster mostly consists of NOTAMs that apply to the entire airport. However, one thing that the red cluster reveals is that NOTAMs that related to “SN REMOVAL” or snow removal and had the words “AP AD” were grouped into this cluster instead of with the other weather NOTAMs. The grouping of weather NOTAMs consists mostly of wet, snow, and ice conditions on runways and taxiways. The green cluster mostly consists of two distinct types of NOTAMs. The first are NOTAMs that refer to “OBST TOWER LGT” or a tower light obstruction. The second are airspace NOTAMs that refer to obstacles in the airspace such as Unmanned Aerial Systems (UAS) or Pyrotechnic Demonstrations. These airspace NOTAMs are different and have a completely different structure compared to the airspace NOTAMs that refer to special use airspace or MOA which had their own cluster in the initial clustering. These two NOTAM types were probably grouped into the same category due to similar words as both try to describe a location of an obstruction mostly with latitude and longitude as well as an altitude. The purple cluster mostly consists of NOTAMs that say “OUT OF SERVICE.” This could refer to runways, taxiways, navigation, or entire airports/aerodromes. Finally, the remaining runway, taxiway, and apron NOTAMs that do not contain the phrase “OUT OF SERVICE” are in the red cluster.

2. Clustering Remarks

While initial clustering on the TF-IDF matrix was able to divide the NOTAMs into overarching categories, sub-clustering was able to group NOTAMs by very specific phrases, topics, or structure. Sub-clustering results were shown for airspace and airport NOTAMs. Sub-clustering results from weather NOTAMs were not as clear and distinct as the results from the two categories displayed. There were some nuanced results which can be saved for later discussion. Sub-clustering helps gain insight into the different types and reasons NOTAMs that are issued and provides one way to separate NOTAMs into many smaller categories of similar NOTAMs. More importantly, this approach shows that NOTAMs can effectively be separated into these smaller categories with a TF-IDF matrix and could pave the way to many more applications such as supervised machine learning algorithms that could automatically categorize new NOTAMs based on TF-IDF scores.

C. Latent Dirichlet Allocation

1. Topic Modelling

According to Kuhn [6], Latent Dirichlet Allocation, or LDA, is a generative probabilistic model that is used for topic modeling. Topic modeling is the task of detecting hidden topics in a set of documents using unsupervised learning. LDA inherently assumes that a document, or NOTAM in this case, is a probability distribution of several global topics, with each topic containing a probability distribution of words.

In this study, scikit-learn’s decomposition module[¶¶] is used to implement LDA to generate 3-topic, 4-topic, 5-topic, and 7-topic models on the NOTAM dataset. Before the LDA models are trained on the NOTAMs, the TF-IDF statistical technique mentioned in the Word Embedding section is used to generate a vector representation, or matrix, for every NOTAM in a designated training set of 100,000 NOTAMs. In addition, certain hyperparameters are tweaked in order to fine-tune the training process. Specifically, the *max_iter* hyperparameter is set to 5, the *random_state* hyperparameter is set to 43, and the *batch_size* hyperparameter is set to 128. The *n_components* hyperparameter is also changed depending on the number of desired topics to be generated for each model. After setting the hyperparameters, the training vectors are passed into each of the LDA models, which generate n-number of topics for the entire training corpus. Initially, every topic contains a temporary assignment of word probabilities, however as the training process proceeds, each LDA model iteratively updates the probabilities corresponding to each word for every topic. As an additional step, the model can also be evaluated after every iteration with the perplexity metric to determine if the predictions are converging. After the total number of iterations specified by the *max_iter* hyperparameter is reached, n-number of topics, each with a final assignment of word probabilities, is outputted.

After the 3-topic, 4-topic, 5-topic, and 7-topic LDA models are trained on the 100,000 NOTAMs, an intertopic distance map, seen in Fig. 7, is used to visualize the semantic distance between the outputted topics for each model. Figure 7 shows the intertopic distance map generated for the 5-topic LDA model and Table 6 depicts the top 10 keywords

[¶¶]`https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html` in each of the 5 topics.

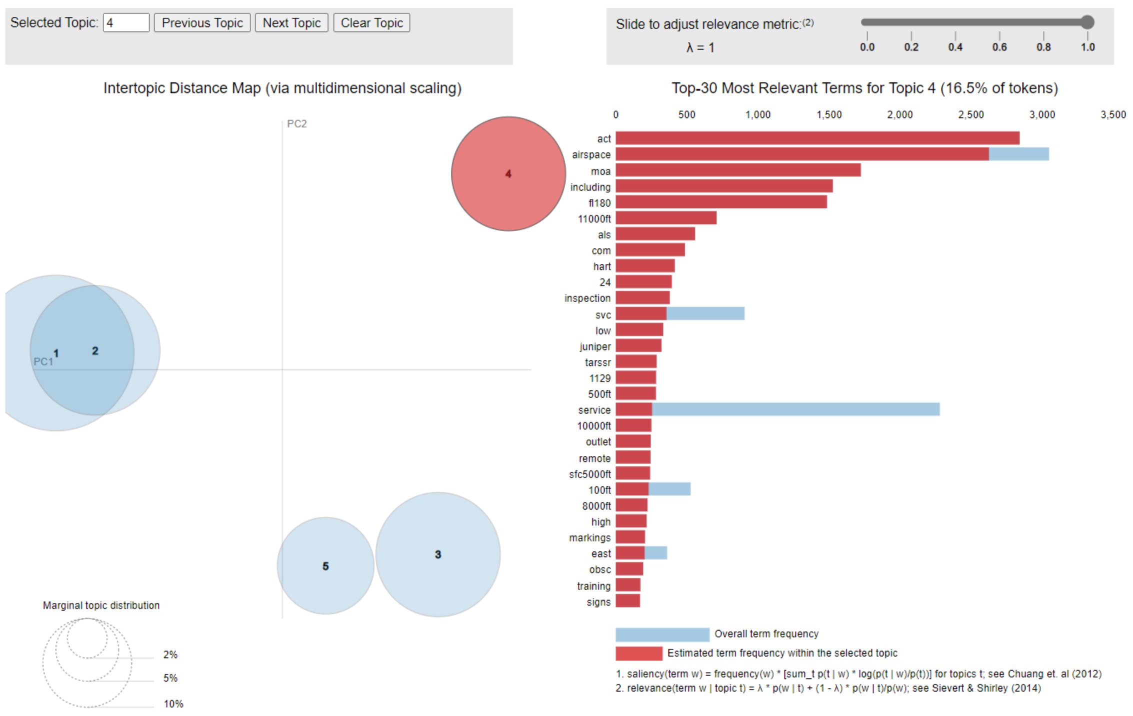

Fig. 7 Intertopic distance map created using pyLDAvis libary for the 5-topic LDA model. `https://pyldavis.readthedocs.io/en/latest/readme.html`

Table 6 Top 10 Keywords for Generated LDA Topics

Topic Number	Subject	Top 10 Words
1	Weather Events on the Taxiway	twy, sn, ficon, obs, dry, apron, clsd, 18in, wet, compacted
2	Weather Events on the Runway	pct, rwy, 100, 555, wet, obs, ficon, nav, ils, service
3	Affected Airport Operations	ap, ad, wip, agl, sfc, obst, lgt, tower, removal, asr
4	Airspace Operations	act, airspace, moa, including, fl180, 11000ft, als, com, hart, 24
5	Affected Runway Operations	rwy, clsd, papi, exc, 32, 25, cond, 13, frost, 28

The distribution of keywords in each topic helps to justify the visualization seen in the intertopic distance map. As seen in Table 6, topic 1 primarily includes NOTAMs that refer to weather conditions on taxiways. This can be seen with the top keywords in topic 1 such as "twy," "sn" which stands for snow, "ficon" which stands for field conditions, "obs," and "dry." Table 6 also shows that topic 2 primarily includes NOTAMs that refer to weather conditions that occur on runways as its top words are "pct," "rwy," "wet," and "ficon." The topic also includes numbers such as 100 and 555. 100 corresponds to the observed snow percent and 555 comes from 5/5/5, which is a runway condition code (RCC) that contains information regarding snow coverage and slipperiness on the runway. As both topic 1 and topic 2 refer to weather conditions and events in areas of an airport, a correlation between the two topics can be expected, which justifies why an overlap between them can be seen on Fig. 7.

The visualization also shows that topics 3 and 5 are relatively close together. Topic 3 includes keywords such as "ad" and "ap," "wip" which stands for work in progress, "agl" which stands for above ground level, "lgt" which stands for light, and "tower." Interestingly, the topic also includes "removal," which is usually used when referring to the removal of snow. This indicates that topic 3 describes airport operations in a general fashion and explains how they are affected by conditions such as weather, which is similar to the findings with k-means clustering. Topic 5 includes keywords such as "rwy," "clsd," "papi" which stands for precision approach path indicator, "cond" which stands for condition, and "frost." Other top keywords include runway numbers such as 32 and 25, which could indicate that operations on these runways are affected by weather. This shows that topic 5 focuses more towards how runway operations are affected by weather conditions and which runways face a closure or change in their operations. This justifies why topics 3 and 5 are in close proximity in Fig. 7 since both topics refer to how operations in areas of the airport, whether in a general or specific manner, are affected by events such as weather.

Finally, topic 4 primarily includes NOTAMs that refer to airspace operations, which can be seen with keywords such as "act" which stands for activated, "airspace," "moa," as well as "fl180" and "11000ft," which indicate altitudes. This explains why topic 4 is located the furthest from any other topic in Fig. 7 as it solely focuses on airspace operations while other topics are more concerned about weather and affected operations in airports.

These results show that the 5-topic LDA model is effective at identifying distinguishable topics. When performing similar analysis on the 3-topic and 7-topic model, it was found that these topic models were not as successful in determining clusters among the data; the 3-topic model did not include topics regarding airport operations besides those on the runway and taxiway and the 7-topic model showed repetition among certain topics. Although the 4-topic model was able to identify distinguishable clusters, these clusters were not as meaningful. From this, it can be concluded that 5 is an optimal number for the *n_components* hyperparameter to get an accurate picture of the different categories of data present in the 100,000 NOTAMs.

2. Topic Prediction

After analyzing the topic distributions to determining that 5 is the optimal number of topics, the 5-topic LDA model is used to generate predictions for the 1,000 NOTAMs allocated for testing. To do this, first, all of the test NOTAMs are vectorized using TF-IDF. Then, they are passed into the 5-topic LDA model, which generates a vector for each NOTAM containing a list of probabilities. Each probability corresponds to the likelihood that a NOTAM belongs to a certain topic. The maximum probability in an outputted vector is used to determine the dominant topic for a NOTAM. Figure 8 represents a process-flow diagram for making predictions with LDA, and Fig. 9 shows a prediction for a test NOTAM made by the 5-topic LDA model.

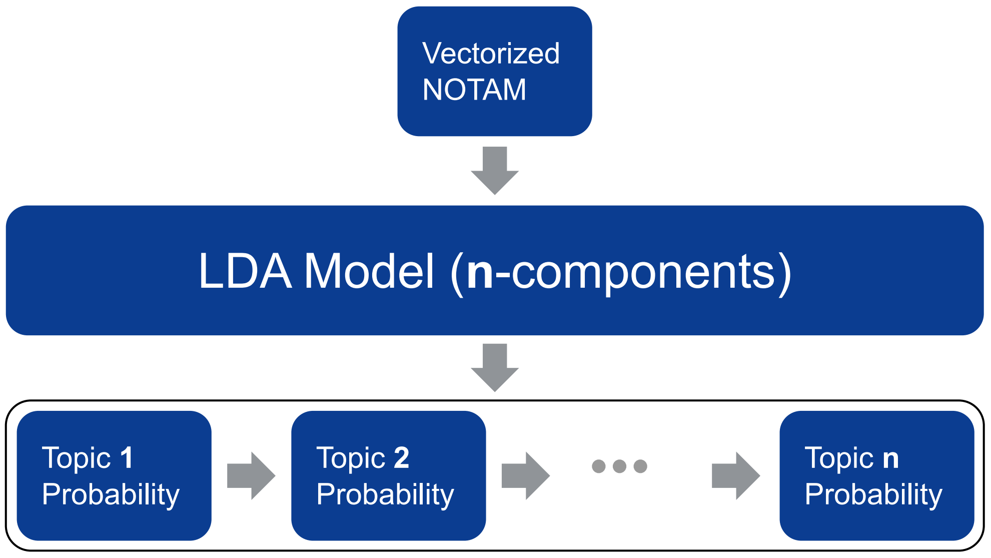

Fig. 8 Process-flow diagram for making predictions with LDA.

As seen in Fig. 9, the test NOTAM—comprising of keywords such as "airspace," "moa," "act," "11000ft," "including," and "fl180"—is predicted by LDA to have topic 4 or Airspace Operations as its dominant topic. When looking at the keywords in topic 4, which contain many of those in the test NOTAM, this prediction makes sense. Therefore, this demonstrates that the 5-topic LDA model is also effective at predicting the category that a NOTAM in the dataset most likely falls under.

Test NOTAM: !SUAW 02/055 ZTL AIRSPACE SNOWBIRD MOA ACT 11000FT UP TO BUT NOT INCLUDING FL180 1902011700-1902011800

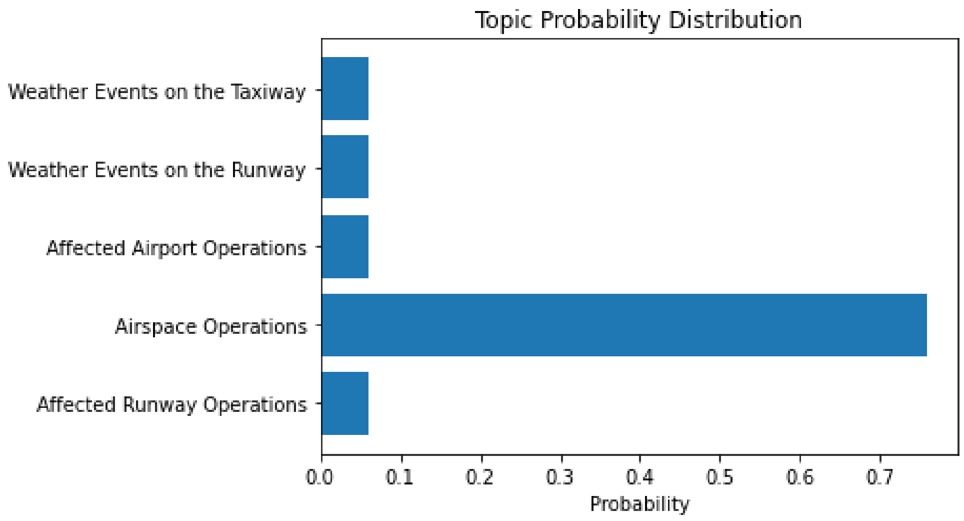

Fig. 9 Predicted topic distribution by a 5-topic LDA model for a test NOTAM.

3. Similarity Detection

Another application of LDA in regards to NOTAMs is similarity detection. Specifically, LDA is used to order a list of NOTAMs based on their similarity to a specific reference NOTAM not part of the list. To accomplish this, first, LDA is used to generate a probability vector for the reference NOTAM and all the NOTAMs in a given list. A probability vector for a NOTAM represents its predicted topic distribution, and therefore, contains details regarding its subject matter. The next step is to calculate the distance between each NOTAM in the list and the reference NOTAM. To calculate the semantic distance between NOTAMs represented as LDA vectors, we use the SciPy implementation of **cosine distance** metric[***]. A smaller cosine distance indicates greater similarity and vice versa. The last step is to order the NOTAMs in ascending order to arrange them from being most similar to least similar to the reference NOTAM.

Figure 10 provides a visual for the similarity predictions for a list of NOTAMs given a reference NOTAM using the 5-topic LDA model. The test NOTAM used in this example is the NOTAM for which topic predictions were done in Fig. 9 to determine that its dominant subject is Airspace Operations. Based on the graph, it can be seen that the NOTAMs that have the smallest log(cosine distance) are those that also refer to operations in the airspace. Meanwhile, the NOTAMs in the list that have the largest log(cosine distance) are those related to weather conditions in the airport. This indicates that the 5-topic LDA model is effective at generating vectors for NOTAMs that encompass their subject matter, allowing for metrics such as cosine distance to be used to accurately identify how similar a NOTAM is to another NOTAM. The ability to perform similarity detection can especially be useful when the LDA model is unable to predict a clear dominant topic for a NOTAM. In such a situation, the generated LDA vector for that NOTAM can be compared using the cosine distance metric with the LDA vectors for NOTAMs that have the conflicting topics as their dominant topics. The topic for which the greatest similarity score is obtained based on the cosine metric can be predicted to be the dominant topic for the NOTAM. Using LDA in conjunction with similarity metrics to categorize data has already been done with great success [34, 35], showing that this is a viable method of making topic predictions.

4. Anomaly Detection

Another application of LDA that is explored is anomaly detection, specifically the usage of LDA to detect the most anomalous NOTAM among a small group of NOTAMs. The research done by Mahapatra, Amogh et al. demonstrates the scalability of LDA in contextual anomaly detection in text data [36]. In their research, LDA was implemented to generate n-number of topics for text logs, which are ranked using symmetrized KL (Kullback-Leibler) divergence, a measure of distance between two probability distributions. Topics ranked above a certain threshold are considered to

[***]`https://docs.SciPy.org/doc/SciPy/reference/generated/SciPy.spatial.distance.cosine.html`

Test NOTAM: !SUAW 02/055 ZTL AIRSPACE SNOWBIRD MOA ACT 11000FT UP TO BUT NOT INCLUDING FL180 1902011700-1902011800

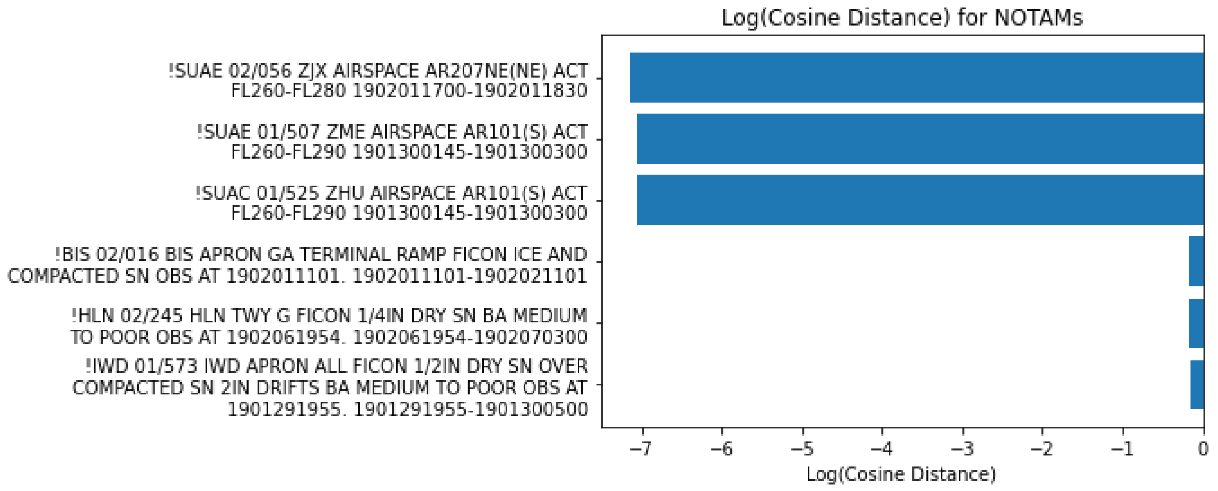

Fig. 10 Predictions by the 5-topic LDA model on the similarity between a reference NOTAM and each of the NOTAMs in a given list.

be normal. Topics ranked below this threshold represent potential anomalies. In order to obtain a more specific and narrowed down list of anomalous topics, an anomaly score is calculated for each of the low-ranked topics. The first step in calculating a score for a low-ranked topic is to sum the similarities between a word in that topic and each word in a normal topic. The second step is to repeat the first step over all words in the low-ranked topic and all topics in the list of normal topics. In the Mahapatra, Amogh et al. study, Normalized Google Distance (NGD) and WordNet are used to detect similarity [36]. After the scoring process is complete, the low-ranked topics are sorted in ascending order based on their score and the topics with the k lowest scores are reported as anomalous.

This study makes use of similar methods as Mahapatra, Amogh et al. to detect anomalies in text data, however there are some differences. As this study aims to understand anomaly detection among the documents themselves and not the topics, an additional step of using LDA to generate a probability vector for each NOTAM is implemented. In addition, as NOTAMs are written in a unique language that is not present in commonly searched text [24], NGD and WordNet cannot be used to evaluate similarity. Instead, SciPy's implementation of the cosine distance metric[†††] is used to determine the semantic distance between two NOTAMs represented as LDA vectors. The next step, similar to Mahapatra, Amogh et al.'s implementation, is calculating a score for every NOTAM in the list. This is done by summing the distances between an individual NOTAM and the other NOTAMs in the group, and then repeating for all instances to get a list of scores. The NOTAMs are then sorted in descending order based on their calculated score. The NOTAM with the highest score is considered to be the most anomalous in the group.

Figure 11 shows the anomaly predictions made for a list of NOTAMs using the 5-topic LDA model. As seen in this example, the four NOTAMs with the lowest anomaly scores relate to airport operations, whether they are on the runway or taxiway, while the NOTAM with the highest anomaly score refers to an airspace operation. This prediction makes sense, as the NOTAM with the highest score was the only NOTAM in the list that depicted changes in the airspace, causing it to be the most anomalous out of all of the five notices. This indicates once again that LDA can effectively capture a NOTAM's subject matter, allowing for anomaly detection operations to be accurately performed.

5. LDA Conclusion

These results show that the LDA statistical technique can be scaled to detect hidden global topics in a NOTAM dataset. They also demonstrate that a trained LDA model can be used to effectively make predictions regarding a NOTAM's subject matter, which can further lead to other end-user applications such as similarity and anomaly comparisons among a group of NOTAMs.

[†††]`https://docs.SciPy.org/doc/SciPy/reference/generated/SciPy.spatial.distance.cosine.html`

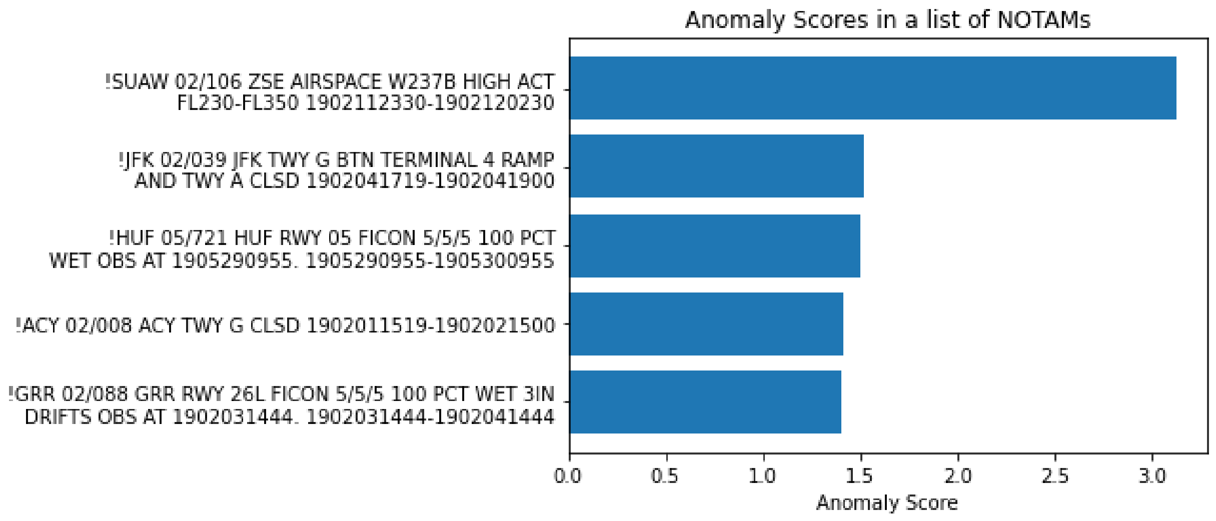

Fig. 11 Predictions by the 5-topic LDA model on the anomaly scores for a list of NOTAMs.

V. Creating a Deeply Parameterized Dataset

Creating a dataset parameterized by different features in a NOTAM could be the start to understanding the structure behind hand-written NOTAMs. Having an understanding of this structure could lead to better search engines or more business rules for models like AIXM.

A single hand-written NOTAM contains one feature, the text that makes up the NOTAM. Certain attributes, such as accountable/effective locations, NOTAM number, keyword, start, and end time, are all easily extractable using string pattern matching, resulting in 7 features. However, even these are limited in the amount of information they give regarding a NOTAM's body. Recalling Fig. 2, NOTAMs can be separated into 16 features. One notable section listed is the condition. This ranges from field conditions (FICON) for surfaces, to CLSD or OUT OF SERVICE for other scenarios. The goal in the next section is to explain a methodology for extracting these features.

A. Named Entity Recognition

Named entity recognition, or NER for short, is a supervised natural language processing task that identifies keywords within a document. In a more traditional approach, if a task required identifying all the airports within a document, then a trained NER model would identify and highlight airports throughout the document (i.e., LAX would be tagged as an airport by the model in the sentence "The aircraft departed LAX").

According to Döhling and Leser [37], NER can also be used to extract patterns within text. That means a well-trained NER model could find the patterns that distinguish each segment within a NOTAM. Therefore, each one of the structure segments described in Fig. 2 can be unique keywords fed into a model and there will be a total of 16 different tags (see Fig. 2 for details). These 16 tags will later be features within a dataset, each representing their respective part of the NOTAM. As an example (see NOTAM that follows), the token **FADED** would be classified as a **Condition** feature according to our model.

!EWR 01/020 EWR TWY EE HLDG PSN MARKINGS FOR ILS BTN RWY 04R/22L AND TWY M **FADED** 2101041504-2106302300

Given that this methodology follows the structure in Fig. 2, FDC NOTAMs are excluded, as they contain information outside of these 16 features.

B. Creating Annotations

Since NER is a supervised machine learning task, a corpus of annotated data must be created. For this task, an open-source annotation tool called Doccano[‡‡‡] is used. Doccano makes the task of annotating language data a more

[‡‡‡]`https://github.com/doccano/doccano`

streamlined process for the user, as well as being a free tool that is easily accessible. Doccano offers many tools for different NLP tasks, but in this case, the entity tagging tool is used. The entity tagging tool was created specifically for the NER task. The software offers ways of creating new tags, tagging highlighted text with hotkeys, and importing/exporting data in multiple different formats.

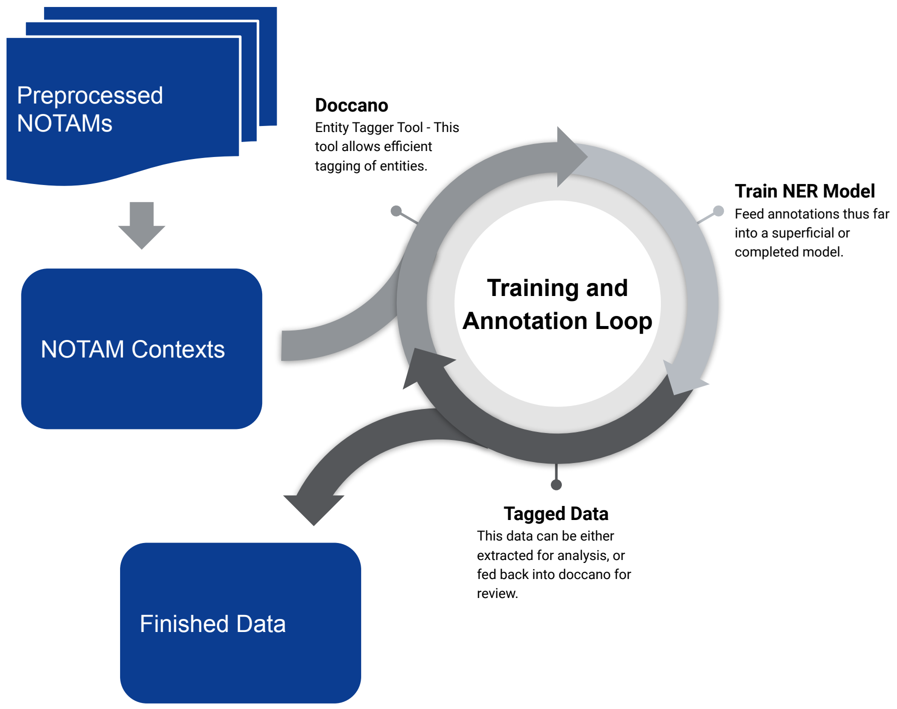
Preprocessed NOTAMs
NOTAM Contexts
Finished Data
Doccano
Entity Tagger Tool - This tool allows efficient tagging of entities.
Training and Annotation Loop
Train NER Model
Feed annotations thus far into a superficial or completed model.
Tagged Data
This data can be either extracted for analysis, or fed back into doccano for review.

Fig. 12 NER Tagging Pipeline Using Doccano.

Figure 12 shows the method used for tagging entities. These NOTAMs are formatted to be imported into Doccano. Then, a small batch of 100-250 NOTAMs are randomly picked from the entire NOTAM dataset. This small batch is manually labeled using Doccano. Using this small batch as a training set, a superficial NER model is trained. This weak model is then used to pre-tag a different batch of 100-250 NOTAMs. With these pre-tagged NOTAMs, the annotator manually checks and fixes any mistakes the weak model has made. After fixing, the new batch is combined with the previous to train a slightly stronger model. This cycle continues, and the task of annotating becomes easier as the developing model trains on more data. This cycle stops when the model is trained to an acceptable F1-score, the metric used to gauge performance. Although this cycle is not guaranteed to result in an acceptable F1-score, an empirical observation after the first few batches suggested improving model performance with increasing data. Currently, 3000 NOTAMs are annotated. This resulting corpus of annotated NOTAMs is now ready for final training and creation of a parameterized dataset.

C. Training

On first thought, using a large pre-trained transformer model, like BERT, to perform NER would be ideal since it would require less training data and contains previously learned representations of words. However, since the NOTAM language is a unique language [24], the previously learned representations on English appeared to make the model perform worse. This could be a result of having to relearn new representations for the abbreviations and acronyms which may have different meaning in general contexts. Because this model performs poorly on the task, a smaller yet still accomplished model architecture by spaCy [38] is used. The spaCy tool offers multiple machine learning models for different NLP tasks, however this section is focused on its NER deep convolutional neural network. Some attributes that spaCy's model offers that align with our task are its strong ability to recognize short entities [38] (entities that do not span multiple words or sentences) which populate most of the entities found in the NOTAMs.

D. Metrics

In order to evaluate the NER model three metrics are computed. Precision, as defined in Eq. 2, tells us what proportion of the positively classified tags are true positives. This tells us how often our prediction method is correct. [39]. Recall (Eq. 3) tells us what proportion of the true positive tags are correctly classified. Since the two metrics are independent of each other and complementary, it is difficult to guarantee a high degree of precision and perfect recall. For example, one can achieve perfect recall by classifying everything as true tags. This of course would lead to poor precision. As a result, a common approach is to use the F1-score (4), which is the harmonic mean between precision and recall, instead. [39].

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3}$$

$$F1\text{-}score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

E. Results

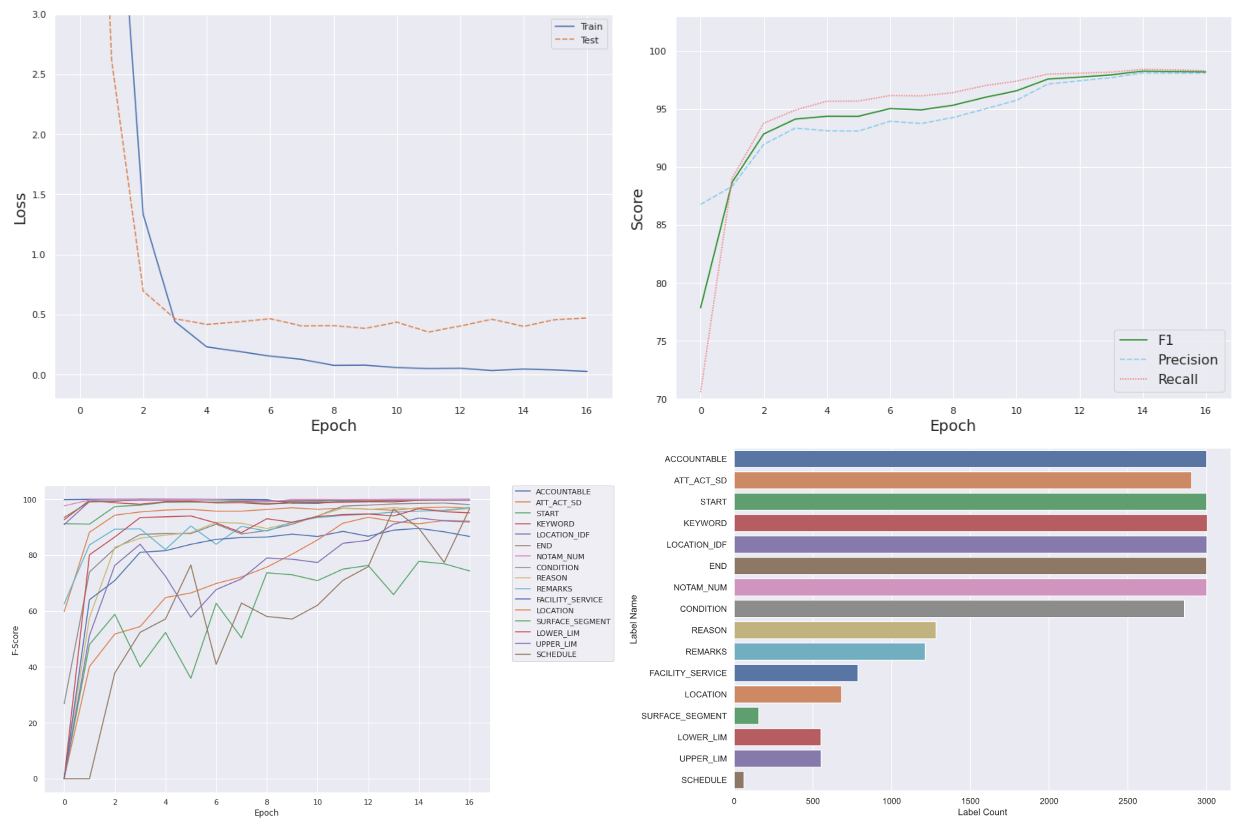

Fig. 13 Named entity recognition testing metrics.

Figure 13 shows the various metrics while training the spaCy NER model. The top left shows loss. This loss is calculated internally by spaCy. This loss was created to optimize whole entity accuracy [§§§]. Loss is analyzed between training and testing sets to understand if the model is under or overfitting. This loss shows a good fit after the first couple of epochs, and does not diverge from the training loss significantly. The top-right shows three performance

[§§§]`https://github.com/explosion/spaCy/blob/master/spacy/pipeline/ner.pyx` metrics mentioned in the previous section. The model has a steep learning curve after the first few epochs, but then starts to level off at epoch 10, settling at around 97-98 for each metric. Interestingly, the precision and recall converge together at these higher epochs as well. The bottom left image shows F1-score of each individual tag. These tags are analyzed individually to show where improvements can be made in the model, and identify individual tags that may bring down the performance of the overall system. As seen by the distribution of label tags on the bottom right, tags that did not have as many occurrences in the annotated corpus performed worse than those labels which occurred commonly. However, there was still strong performance in even rarer tags with nearly 0.78 F1-score for the SURFACE_SEGMENT tag which only contains 156 occurrences (5.2% of all NOTAMs).

Table 7 Parameterized NOTAM Dataset

Accountable	NOTAM Number	Location Identifier	Keyword	...	Condition	Reason	...	Start	End
HON[¶¶¶]	04/190	HON	AD	...	CLSD	NaN	...	1904271015	1904272200
CYS[17]	10/052	CYS	RWY	...	FICON	10 PCT 1/8IN WET SN...	...	1904271015	1904272200
SUAW[18]	10/708	ZSE	AIRSPACE	...	ACT	NaN	...	1904271015	1904272200
CMI[19]	01/644	CMI	TWY	...	FICON	1IN DRY SN BA MEDIUM	...	1904271015	1904272200

Table 7 shows some examples of the resulting NER dataset. As described earlier, each NOTAM is successfully segmented into 16 different sections (only 7 shown in Table 7). Note that not all NOTAMs have the same structure, and may not contain every segment. These instances are filled as "NaN" in the dataset. Refer back to Fig. 13 (bottom-left) for the distribution of labels in the training data. This label distribution also holds true with the final tagged dataset, as seen in Fig. 14. The sankey diagram in Fig 14 shows the distribution and structural flow of the tagged dataset. Each vertical grey bar represents a column in the dataset, while the connections are colored by the "KEYWORD" column to analyze different structures based on the type of NOTAM. As shown, NOTAM types such as RWY, TWY, and APRON have a similar structure. This is thought to be primarily dominated by weather NOTAMs, as they typically follow the structure:

$$\text{KEYWORD} \longrightarrow \text{ATT\_ACT\_SD} \longrightarrow \text{CONDITION} \longrightarrow \text{REASON} \longrightarrow \text{REMARKS}$$

Another notable structure are AIRSPACE NOTAMs which typically follow the structure:

$$\text{KEYWORD} \longrightarrow \text{FACILITY\_SERVICE} \longrightarrow \text{LOWER\_LIM} \longrightarrow \text{UPPER\_LIM} \longrightarrow \text{CONDITION}$$

As a final remark, this solution is not the only way to extract specific information from NOTAMs. Digital NOTAMs contain most, if not all, of this information in a formally structured XML format which is easy to query. Where this method outperforms that solution is for NOTAMs that are not captured digitally (recall: 30% of NOTAMs are not captured digitally). An example are AIRSPACE NOTAMs, which do not have specification by the digital NOTAM format. They are a large portion that can now be partially explained by our proposed NER model.

Another use-case of our model is converting hand-written NOTAMs into the digital format. NOTAMs that follow the digital NOTAM specification but were not created in *NOTAM Manager* are not recognized as digital NOTAMs, and do not have the added benefits such as better querying and NOTAM format translation. Although our current model does not recognize all of the scenarios posed by the digital NOTAM specification, it can bee seen as the first step in outlining NOTAMs, before even more detailed information can be extracted and formatted into a formal digital NOTAM.

VI. Question and Answering Using Plain Language NOTAMs

Another application to NOTAMs explored is performing the *Question and Answering* task. In aviation, Q&A is not new, and has been utilized by IBM's Watson for general aviation service and maintenance [14]. However, this task has not been utilized as much in air traffic management documents. Q&A could prove useful in querying information from data if it is faster than a human reading and interpreting the data, and retrieves information that is not easily obtainable using simple parsing techniques [40]. This makes NOTAMs a good candidate for the task since consumers of NOTAMs may need to read through dozens to hundreds of NOTAMs to get the entire details of an airport or airspace condition.

[¶¶¶]!HON 04/190 HON AD AP CLSD 1904271015-1904272200

[17]!CYS 10/052 CYS RWY 27 FICON 10 PCT 1/8IN WET SN PLOWED AND SWEPT 100FT WID OBS AT 1910271928. 1910271928-1910281928

[18]!SUAW 10/708 ZSE AIRSPACE R6714C ACT SFC-15000FT 1910201300-1910211300

[19]!CMI 01/644 CMI TWY D FICON 1IN DRY SN BA MEDIUM OBS AT 1901270950. 1901270950-1901280950

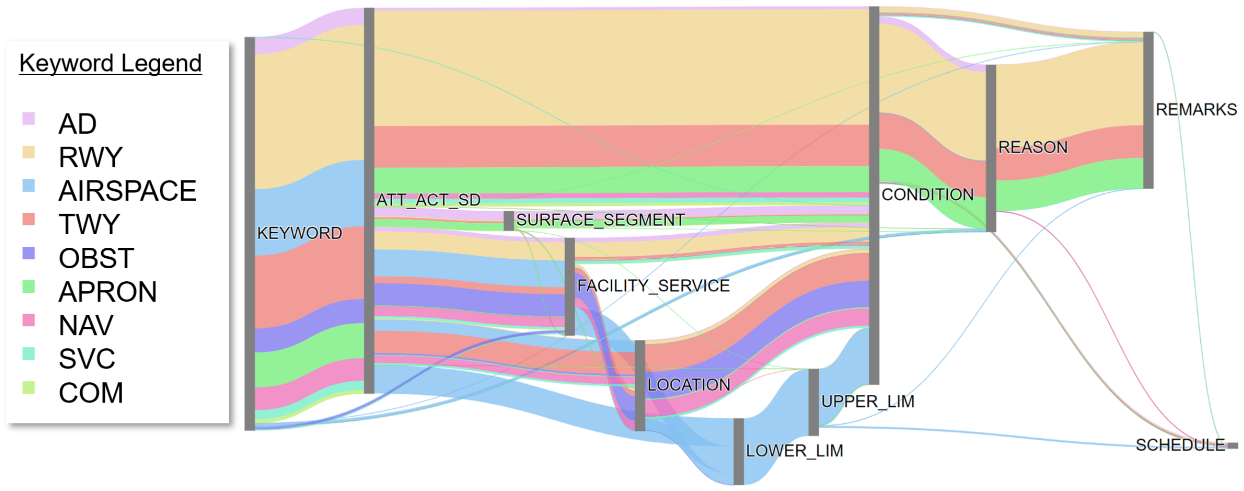

Fig. 14 Sankey diagram: resulting NOTAM structure. Sampled from one million tagged NOTAMs.

The other benefit is that NOTAMs can be free-form, containing enough variation that simple parsing techniques do not work as well as machine learning techniques could.

One of the first problems encountered with the NOTAM data is that NOTAMs are written in a special highly-abbreviated language and there is no way to ask questions within this language. Gupta et al. [41] would classify the NOTAM language as a *resource-scarce* language as opposed to a *resource-rich* language like English. Therefore the solution would be to translate NOTAMs into English and utilize the tools English offers to ask meaningful questions. This also allows us to reap the benefits of transformer models that are pretrained on large English corpora. Luckily, the *FAA Plain Language Program*[20] pushes aviation to offer plain language alternatives to all domain-knowledge heavy documents, including NOTAMs. Because of this effort, AIXM digital NOTAM format supports the conversion of digital NOTAMs into a plain language format. The downside is that AIXM does not capture all types of NOTAMs (as reiterated many times). The original 3.7 million NOTAM dataset only contains a little over one million plain language NOTAMs.

Q&A task is not necessarily a machine learning task. It could involve employing simple search engines to find answers, use machine learning techniques, retrieve from an ontology and/or a mixture of all of the above [9]. However, in this work, we focus on a machine learning approach because of the recent rise in deep transformer models. Q&A also has a few different forms in machine learning. The specific type of Q&A being done in this research is extractive Q&A, where the inputs to each model will be question and context pairs, while the outputs are the positional indices of where the start and end of the answer are within the text. Figure 15 shows an example of how this extraction works. Both the context and question are inputted to the model, and the model will return two lists of scores for both the starting and ending position of possible answers.

Some downsides to this method are that the models cannot synthesize answers that do not appear in the text itself, and the models *must* return an answer, even if there is no relevant answer to the question in the context provided. In future work, we plan to build upon this research and explore variations of the Q&A tasks such as *hybrid Q&A model* that utilizes both knowledge graphs along with deep learning techniques. These systems are useful as they allow the speed and accuracy of a knowledge graph, with the benefit of free text/speech which is prevalent in the domain. Also, this route is extremely viable since most NOTAMs are already modeled by a knowledge graph in the AIXM.

Three popular transformer models are studied and compared for their effectiveness in understanding the task: Bidirectional Encoders from Representations (BERT) [8], Robustly optimized BERT Pretraining Approach (RoBERTa) [42], and XLNet, a generalized autoregressive pretraining method for language understanding [43]. Each model is trained and validated on identical data to find differences in performance and gauge their usefulness within NOTAM data and possibly the broader air traffic management domain.

[20]`https://www.faa.gov/about/initiatives/plain_language/`

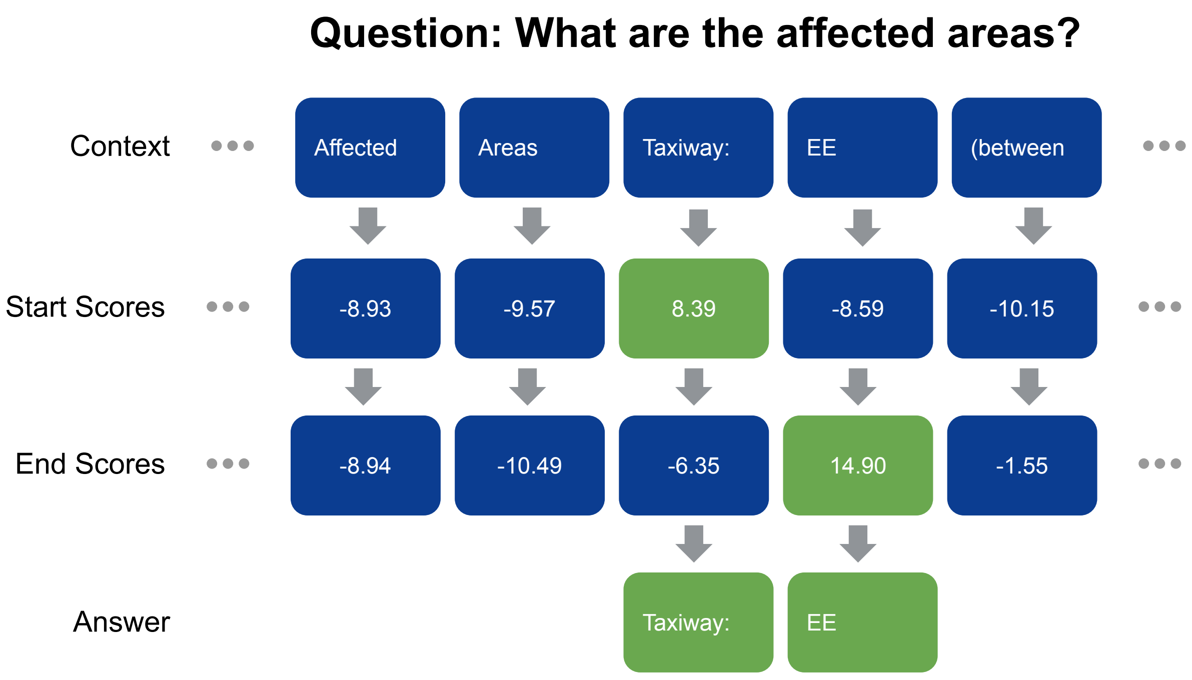

Fig. 15 QA Extraction Example.

Note: This example was created using XLNet-Large, however the tokenization was compressed for readability (in reality special characters and words will be broken up more).

A. Fine-tuning of Transformer Models

Each transformer model looked at has some version of unsupervised fine-tuning step which is used to initialize weights of the model to particular data before training on a downstream task such as Q&A. BERT uses masked language modeling (MLM) and next-sentence prediction [8] and RoBERTa uses dynamic MLM [42]. MLM is a *fill in the blank* task, where a model uses the context words surrounding a mask token to try to predict what the masked word should be. For an input that contains one or more mask tokens, the model will generate the most likely substitution for each [8]. A difference between BERT and RoBERTa is that RoBERTa randomizes masks between epochs during training [42] (thus the word ‘dynamic’) whereas BERT masks the labels once at the beginning of training. XLNet is slightly different and uses permutation language modeling [43]. Although the outputs of all three models are the same, XLNet predicts the masked input by calculating the output based on all permutations of the input sequence. This introduces bidirectionality to an otherwise unidirectional model [43].

B. Preparing Context, Question, and Answering Pairs

We list the various methods used to create context-question-answer triplets below.

1) Manual annotation: this simple but tedious way is to have a human type out questions and answers given a NOTAM and repeat this process thousands of times.
2) Generic question generation: this slightly less tedious method generates very common questions that could be answered by reading any type of NOTAM. Some examples are “What is the NOTAM number?” or “What is the affected area?” These questions can be asked many times, requiring the annotator to simply highlight the answer instead of thinking about a more specific question to write down.
3) Context generation: another automated question technique that requires altering or synthesizing new NOTAMs that have slightly different contents, but the same question-answer structure. An example would be regarding the question “How much snow coverage is on the runway?” to the answer “100 PCT SNOW.” The context surrounding the answer could be changed, such as the airport and runway pair. The answer could also be altered

to be "50 PCT SNOW" while keeping the context the same, or performing all of these together.

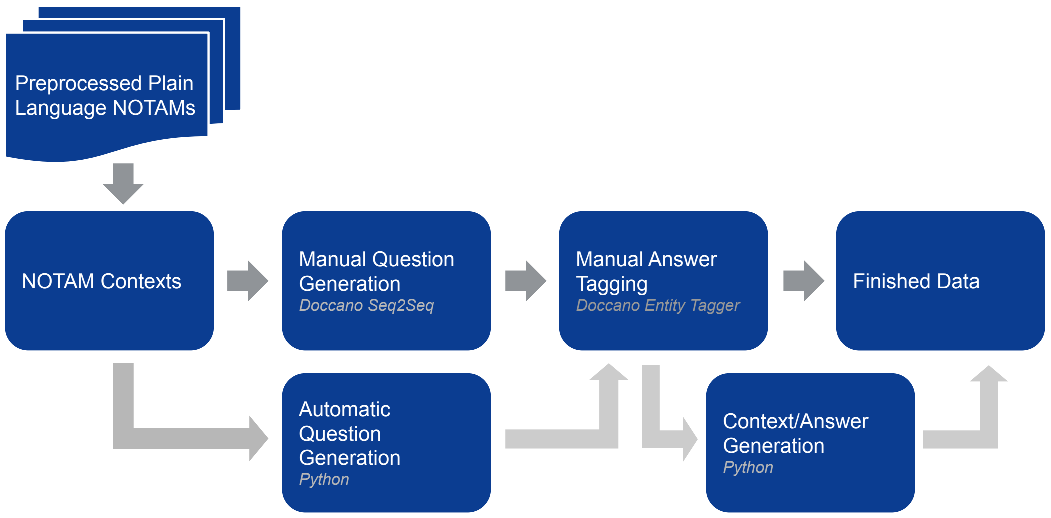
Preprocessed Plain Language NOTAMs
NOTAM Contexts
Manual Question Generation
Doccano Seq2Seq
Manual Answer Tagging
Doccano Entity Tagger
Finished Data
Automatic Question Generation
Python
Context/Answer Generation
Python

Fig. 16 Question and answer annotation pipeline.

For the manual annotations, Doccano (as mentioned earlier) was used. Instead of using just the entity tagger, the *sequence-to-sequence* feature was used in tandem with the entity tagger to create the annotations. An overview of the workflow is displayed in Fig. 16. First, preprocessed plain language NOTAMs are imported into Doccano or Python. Doccano is used to start the manual annotation process by displaying the contexts one by one and awaits new questions in response. In Python, automated questions may be generated as mentioned above. These new question-context pairs are then taking into Doccano for manual answer tagging. After manual answer tagging, some NOTAMs are then taken for analysis of automatic context-question-answer generation. Those who follow similar format as mentioned earlier will be duplicated and modified as *context generation* entries.

C. Metrics

Three common metrics used to evaluate the Q&A task are accuracy, exact match, and F1-score [9]. Accuracy, as calculated in Eq. 5 gives a sense of how well the model is picking out relevant information. However, one problem in Q&A systems is the large number of true negatives [9] since the correct answers are typically a small subset of the larger context surrounding them. Exact match is evaluated simply by checking if the answer exactly matches the expected result (Eq. 6). While this gives us an idea of how our model is performing, it has the issue of giving many false-negatives since any partially correct answer is flagged as wrong. Finally, F1-score (Eq. 4) is used in a manner similar to named entity recognition. As F1-score includes information about both recall and precision of each token in the answer, it penalizes the model less for not getting the exact answer, but still recognizes partial answers as being somewhat correct. Figure 17 gives an example of how true positive, true negative, false positive, and false negatives are calculated in this context.

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + True\ Negative + False\ Negative} \tag{5}$$

$$Exact\ Match = \begin{cases} 1, & \text{if } Prediction = Truth \\ 0, & \text{if } Prediction \neq Truth \end{cases} \tag{6}$$

D. Results and Comparison

Table 8 gives the results of training all transformer models. Each model was trained on an NVIDIA Tesla P100 using their default configurations from Hugging Face[21] with the exception of epochs, which is fixed at 10 for Q&A and

[21]`https://huggingface.co/`

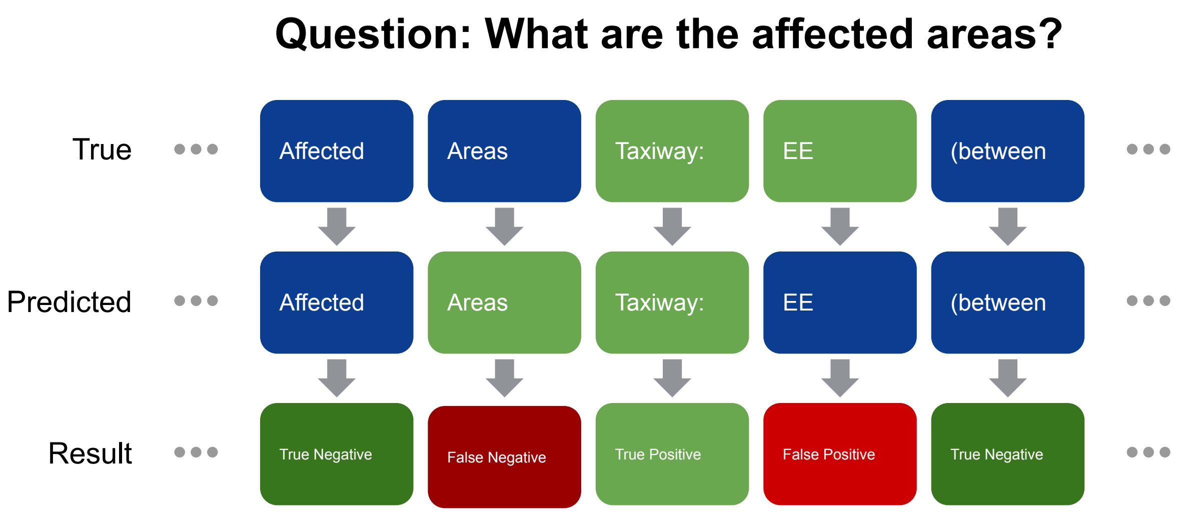

Fig. 17 NLP Classification Metrics

1 for fine-tuning, and batch size, which is fixed at 8 for the base cases and 4 for the large cases. Unfortunately, our hardware did not allow any higher batch size without running into memory issues.

Table 8 Transformer Model Validation Results

Model	LM Exact Match	Q&A Accuracy	Q&A Exact Match	Q&A F1-score
BERT-Base-Uncased	–	0.908	0.429	0.542
BERT-Base-Uncased (Fine-tuned)	0.879	0.939	0.546	0.724
BERT-Large-Uncased	–	0.962	0.699	0.826
BERT-Large-Uncased (Fine-tuned)	0.885	0.945	0.576	0.712
RoBERTa-Base	–	0.972	0.784	0.874
RoBERTa-Base (Fine-tuned)	0.890*	0.968	0.767	0.844
RoBERTa-Large	–	0.976*	0.794*	0.883
RoBERTa-Large (Fine-tuned)	0.881	0.970	0.759	0.870
XLNet-Base-Cased	–	0.974	0.769	0.884
XLNet-Large-Cased	–	0.976*	0.772	0.889*

Note: Each cell includes numbers reported from the best epoch of training. Scores labeled with a * symbol represent the best score for a given metric.

Fine-tuning consisted of performing each model's respective fine-tuning task on 200,000 NOTAMs for training and 20,000 NOTAMs for testing. More fine-tuning data is possible in the future with better hardware. The final annotated Q&A dataset consisted of 4,015 question-answer-context triplets, which was split 90-10 for training and testing (3,614 train, 401 test).

The best results for each metric are highlighted with a * symbol. We observed that RoBERTa-Large and XLNet-Large-Cased performed best in *accuracy*, RoBERTa-Large performed best in *exact match*, and XLNet has the best F1-score. These results align relatively well with what was expected, with the larger models tending to outperform the smaller ones. Where the results misaligned with our expectations are the results of the fine-tuned models. We expected that fine-tuning each model on a large number of NOTAMs would help the models get a deeper understanding of the structure of each plain language NOTAM. However, as the results show, every model other than BERT-Base-Uncased took a hit in performance. One possible explanation of this trend could be the number of training examples fed into the model. Due to time and hardware constraints, only 200,000 out of 3.7 million NOTAMs were used for fine-tuning. If more data were made available for fine-tuning, we expect to see an improvement over the base models.

Table 9 Q&A Examples

Context NOTAM		
Issuing Airport: (EWR) Newark Liberty Intl		
NOTAM Number: 01/020		
Effective Time Frame		
Beginning: Monday, January 4, 2021 1504 (UTC)		
Ending: Wednesday, June 30, 2021 2300 (UTC)		
Affected Areas		
Taxiway: EE (between RWY 04R/22L and TWY M)		
Marking Type: Holding position markings for ILS		
Status: Faded		
Model	Answer	Score
Question: What is the affected area?		
BERT-Large-Uncased	Taxiway: EE	0.413
RoBERTa-Large	Taxiway: EE	0.999
XLNet-Large-Cased	Taxiway: EE	0.687
Question: What is the status of the holding position markings?		
BERT-Large-Uncased	Faded	0.979
RoBERTa-Large	Faded	0.999
XLNet-Large-Cased	Faded	0.998
Question: What is the affected airport?		
BERT-Large-Uncased	(EWR) Newark Liberty Intl	0.142
RoBERTa-Large	(EWR) Newark Liberty Intl	0.998
XLNet-Large-Cased	(EWR) Newark Liberty Intl	0.949
Question: Where is taxiway EE closed?		
BERT-Large-Uncased	between RWY 04R/22L and TWY M	0.690
RoBERTa-Large	between RWY 04R/22L and TWY M	0.997
XLNet-Large-Cased	between RWY 04R/22L and TWY M	0.917

Note: Example results of three different trained models.

Table 9 shows some examples of different questions asked to the same context NOTAM. The best of each model is shown: BERT-Large-Uncased, RoBERTa-Large, and XLNet-Large-Cased. Although all models perform well and give correct answers, there are differences in the scores each model gives to the answer.

VII. Summary and Future Research Direction

In this research work, we have investigated the application of supervised and unsupervised machine learning techniques to gain a deeper understanding of the patterns, structure, and content of existing NOTAMs. Some of the unique patterns we discovered can help better classify NOTAMs, detect variations and anomalies among NOTAMs, and help guide advanced NOTAM query systems for individual user or system needs.

In summary, key results and learnings from our work are included below.

1) **Exploratory Data Analysis**
   Using TF-IDF embeddings and k-means clustering, NOTAMs were successfully categorized into three overarching

clusters: Airport (NOTAMs pertaining to Airport conditions), Weather, and Airspace. These broad clusters could be sub-clustered into more specific clusters. Using those same learned embeddings, LDA was used to find hidden topics and demonstrated the ability to make effective predictions of a NOTAM's subject matter and degree of similarity to other NOTAMs.

2) **Creating a Deeply Parameterized Dataset**
Using manually annotated NOTAMs, a NER model was successfully trained to an F1-score of 98%. The tags generated from this process also led to deeper insights regarding the structure of NOTAMs with different contexts.
3) **Question and Answering Using Plain Language NOTAMs**
Along with understanding plain language NOTAMs, we have evaluated the use of deep natural language transformer models within the domain of NOTAMs. We see that both RoBERTa and XLNet are great candidates for performing tasks, especially the question answering task.

While most NOTAMs are captured digitally and understood via a standardized model (AIXM), our effort helps gain an understanding of the NOTAMs not currently digitized. As a future research goal, we plan to build a system that can auto-convert current hand-written NOTAMs into a digital NOTAM specification.

Question and answering has been used in this work to evaluate the performance of modern deep learning models. There are other NLP tasks that can be explored and optimized for use in the context of NOTAMs. One potential future research topic will be to expand extractive question answering to more generalized (applicable) tasks in air traffic management. Performing this Q&A task can be useful, especially when utilizing end-user tools like voice assistants. However, current models can only extract information from one NOTAM at a time. To make the Q&A model more effective, a document-retrieval system could be made that gathers relevant NOTAMs for the Q&A model to extract answers. Additionally, hybrid Q&A tasks mentioned by Soares and Parreiras [9] could be utilized. An extractive Q&A model in conjunction with AIXM could both provide an interface for natural language interaction as well as contain the highest data quality required by the aviation community.

Along with NOTAMs, we hope to apply similar NLP methods to extract insights and patterns from other heritage air traffic management documents such as *Letters of Agreement* (LoAs) and *Standard Operating Procedures* (SOPs). This work will inform and facilitate future NAS operations as the airspace becomes increasing complex with diverse users, diverse operations, and diverse flying systems.

References

[1] "NOTAMs: Back to Basics: Pilots," *Federal Aviation Administration*, 2020. URL `https://www.faasafety.gov/files/notices/2019/Feb/Pilots_NOTAM_primer_1-28-19.pdf`.

[2] "Digital NOTAM," *Aeronautical Information Exchange Model*, Online. URL `https://aixm.aero/page/digital-notam`.

[3] Burgstaller, F., Steiner, D., Neumayr, B., Schrefl, M., and Gringinger, E., "Using a model-driven, knowledge-based approach to cope with complexity in filtering of notices to airmen," *Proceedings of the Australasian Computer Science Week Multiconference*, 2016, pp. 1–10.

[4] Liddy, E. D., "Natural language processing," 2001.

[5] Young, T., Hazarika, D., Poria, S., and Cambria, E., "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine*, Vol. 13, No. 3, 2018, pp. 55–75.

[6] Kuhn, K. D., "Using structural topic modeling to identify latent topics and trends in aviation incident reports," *Transportation Research Part C: Emerging Technologies*, Vol. 87, 2018, pp. 105–122.

[7] Yu, B., and Wei, J., "IDCNN-CRF-based domain named entity recognition method," *2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT*, IEEE, 2020, pp. 542–546.

[8] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423, URL https://www.aclweb.org/anthology/N19-1423.

[9] Calijorne Soares, M. A., and Parreiras, F. S., "A literature review on question answering techniques, paradigms and systems," *Journal of King Saud University - Computer and Information Sciences*, Vol. 32, No. 6, 2020, pp. 635–646. https://doi.org/https://doi.org/10.1016/j.jksuci.2018.08.005, URL https://www.sciencedirect.com/science/article/pii/S131915781830082X.

[10] Di Flumeri, G., De Crescenzio, F., Berberian, B., Ohneiser, O., Kramer, J., Aricò, P., Borghini, G., Babiloni, F., Bagassi, S., and Piastra, S., "Brain–computer interface-based adaptive automation to prevent out-of-the-loop phenomenon in air traffic controllers dealing with highly automated systems," *Frontiers in human neuroscience*, Vol. 13, 2019, p. 296.

[11] Westin, C., "Strategic Conformance: Exploring Acceptance of Individual-Sensitive Automation for Air Traffic Control," Ph.D. thesis, Delft University of Technology, 2017.

[12] Stroup, R. L., and Niewoehner, K. R., "Application of Artificial Intelligence in the National Airspace System–A Primer," *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, IEEE, 2019, pp. 1–14.

[13] Stroup, R. L., Niewoehner, K. R., Apaza, R. D., Mielke, D., and Mäurer, N., "Application of AI in the NAS–the Rationale for AI-Enhanced Airspace Management," *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, IEEE, 2019, pp. 1–10.

[14] Paul, S., Purkaystha, B. S., and Das, P., "NLP TOOLS USED IN CIVIL AVIATION: A SURVEY," *International Journal of Advanced Research in Computer Science*, Vol. 9, 2018, pp. 109–114.

[15] Clark, P., and Harrison, P., "Boeing's NLP system and the challenges of semantic representation," *Semantics in Text Processing. STEP 2008 Conference Proceedings*, 2008, pp. 263–276.

[16] Rose, R. L., Puranik, T. G., and Mavris, D. N., "Natural Language Processing Based Method for Clustering and Analysis of Aviation Safety Narratives," *Aerospace*, Vol. 7, No. 10, 2020, p. 143.

[17] Kierszbaum, S., and Lapasset, L., "Applying Distilled BERT for Question Answering on ASRS Reports," *2020 New Trends in Civil Aviation (NTCA)*, IEEE, 2020, pp. 33–38.

[18] Tanguy, L., Tulechki, N., Urieli, A., Hermann, E., and Raynal, C., "Natural language processing for aviation safety reports: From classification to interactive analysis," *Computers in Industry*, Vol. 78, 2016, pp. 80–95.

[19] Srinivasan, P., Nagarajan, V., and Mahadevan, S., "Mining and classifying aviation accident reports," *AIAA Aviation 2019 Forum*, 2019, p. 2938.

[20] Madeira, T., Melício, R., Valério, D., and Santos, L., "Machine learning and natural language processing for prediction of human factors in aviation incident reports," *Aerospace*, Vol. 8, No. 2, 2021, p. 47.

[21] Xing, Z., Dai, Z., Luo, Q., Liu, Y., Chen, Z., and Wen, T., "Research on Name Entity Recognition Method in Civil Aviation Text," *2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT*, IEEE, 2020, pp. 23–29.

[22] Bravin, M., Pfäffli, D., Mazumder, S., and Pouly, M., "Automated Smartification of Notices to Airmen," *2020 7th Swiss Conference on Data Science (SDS)*, IEEE, 2020, pp. 51–52.

[23] , ????

[24] "What is a NOTAM?" *Federal Aviation Administration*, 2021. URL https://www.faa.gov/about/initiatives/notam/what_is_a_notam/.

[25] "7930.2S - Notices to Airmen (NOTAM)," *Federal Aviation Administration*, 2019. URL https://www.faa.gov/documentLibrary/media/Order/7930.2S_Notices_to_Airmen_(NOTAM).pdf.

[26] Wallace, E., Wang, Y., Li, S., Singh, S., and Gardner, M., "Do NLP Models Know Numbers? Probing Numeracy in Embeddings," *CoRR*, Vol. abs/1909.07940, 2019. URL http://arxiv.org/abs/1909.07940.

[27] Ramos, J., et al., "Using tf-idf to determine word relevance in document queries," *Proceedings of the first instructional conference on machine learning*, Vol. 242, Citeseer, 2003, pp. 29–48.

[28] Zhang, W., Yoshida, T., and Tang, X., "A comparative study of TF* IDF, LSI and multi-words for text classification," *Expert Systems with Applications*, Vol. 38, No. 3, 2011, pp. 2758–2765.

[29] Steinley, D., "K-means clustering: a half-century synthesis," *British Journal of Mathematical and Statistical Psychology*, Vol. 59, No. 1, 2006, pp. 1–34.

[30] Singh, A., Yadav, A., and Rana, A., "K-means with Three different Distance Metrics," *International Journal of Computer Applications*, Vol. 67, No. 10, 2013.

[31] Cao, L., Chua, K. S., Chong, W., Lee, H., and Gu, Q., "A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine," *Neurocomputing*, Vol. 55, No. 1-2, 2003, pp. 321–336.

[32] Van der Maaten, L., and Hinton, G., "Visualizing data using t-SNE." *Journal of machine learning research*, Vol. 9, No. 11, 2008.

[33] McInnes, L., Healy, J., and Melville, J., "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[34] Priyantina, R. A., and Sarno, R., "Sentiment analysis of hotel reviews using Latent Dirichlet Allocation, semantic similarity and LSTM," *Int. J. Intell. Eng. Syst*, Vol. 12, No. 4, 2019, pp. 142–155.

[35] Puspaningrum, A., Siahaan, D., and Fatichah, C., "Mobile app review labeling using lda similarity and term frequency-inverse cluster frequency (TF-ICF)," *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, IEEE, 2018, pp. 365–370.

[36] Mahapatra, A., Srivastava, N., and Srivastava, J., "Contextual Anomaly Detection in Text Data," *Algorithms*, Vol. 5, 2012, pp. 469–489. https://doi.org/10.3390/a5040469.

[37] Döhling, L., and Leser, U., "EquatorNLP : Pattern-based Information Extraction for Disaster Response," 2011.

[38] "spaCy," *Explosion*, Online. URL https://spacy.io/.

[39] Derczynski, L., "Complementarity, F-score, and NLP Evaluation," *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, European Language Resources Association (ELRA), Portorož, Slovenia, 2016, pp. 261–266. URL https://www.aclweb.org/anthology/L16-1040.

[40] Kierszbaum, S., and Lapasset, L., "Applying Distilled BERT for Question Answering on ASRS Reports," *2020 New Trends in Civil Aviation (NTCA)*, 2020, pp. 33–38. https://doi.org/10.23919/NTCA50409.2020.9291241.

[41] Gupta, D., Kumari, S., Ekbal, A., and Bhattacharyya, P., "MMQA: A multi-domain multi-lingual question-answering framework for English and Hindi," *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[42] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V., "Ro{BERT}a: A Robustly Optimized {BERT} Pretraining Approach," , 2020. URL https://openreview.net/forum?id=SyxS0T4tvS.

[43] Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., and Le, Q. V., "XLNet: Generalized Autoregressive Pretraining for Language Understanding," *CoRR*, Vol. abs/1906.08237, 2019. URL http://arxiv.org/abs/1906.08237.