Classification of Notices to Airmen using Natural Language Processing

Aiden C. Szeto[1]
University of California - Los Angeles, Los Angeles, CA, 90095

Aditya Das[2]
National Aeronautics and Space Administration, Moffett Field, CA, 94035

Abstract* – This paper establishes the feasibility of using Natural Language Processing (NLP) to classify NOTAMs or Notices to Airmen – a pilot messaging framework to gather real-time situational awareness. Present day air mobility operations heavily rely on NOTAMs. However, pilots often have difficulty interpreting NOTAMs due to the sheer volume of inapplicable messages and unclear abbreviations. Using NLP, the presented study analyzes the accuracy of classifying NOTAMs and, thereby, the efficiency of generating actionable interpretations in real time. To this effect, efficacies of four NLP neural network architectures were analyzed, including three Recurrent Neural Networks (RNNs) with GloVe, Word2Vec, and FastText word embeddings, and one trained Bi-Directional Encoder Representations from Transformers (BERT) model. The four neural networks were trained and evaluated on three open-source datasets of varying text lengths, vocabularies, and grammars, taken from e-commerce product descriptions, social media tweets, and unstructured descriptions for data and analytics services on open data marketplaces such as NASA's Data and Reasoning Fabric (DRF) platform. This provided cross-analysis of each neural network architecture's performance per text type. The best performing architecture, BERT, was then fine-tuned on a collection of open-source NOTAM data. Post-training, a real-time NOTAM classification service was implemented to draw inference on new NOTAMs using the trained model, which demonstrated close to 99% accuracy in classification. This modular classification service is envisioned to be integrated with a data and analytics delivery platform, such as the DRF, thus availing real-time contextualization of NOTAMs to air mobility clients, humans, and machines for enhanced decision making.

I. Introduction

Notices to Airmen (NOTAMs) are notices containing information essential to personnel concerned with flight operations. NOTAMs are sent by government agencies and airport operators and are used to communicate real-time status updates that are not known early enough to be broadcasted by other means. They indicate abnormal statuses including the establishment, condition, or change of any facility, service, procedure, or hazard of any component of the National Airspace System (NAS) [1].

Because they contain such critical safety information for airspace stakeholders, NOTAMs have adapted a specific seven-line format. They are written completely in upper case and contain various special contractions to make communication more efficient, Fig. 1. However, the lack of standardization, particularly in the United States, along with the convoluted abbreviation system and sheer volume of NOTAMs has caused serious issues in the past. For example, Air Canada Flight 759 nearly crashed into four other airlines as it attempted to land on a San Francisco taxiway in July 2017 because information was not extracted from a NOTAM [2]. In many situations, pilots may not be familiar with all the coding and abbreviations found in a NOTAM – as a result, flight-critical information is missed, and flight personnel safety is jeopardized.

Recently, there have been pushes by the Federal Aviation Administration (FAA) to align United States domestic NOTAMs with International Civil Aviation Organization (ICAO) standards [3]. However, this alone will not eliminate the confusion pilots have to identify critical elements in large briefing packages of coded NOTAMs. While recent digitalization of NOTAMs has helped improved organization, there are significant strides to be made improving this message system for pilots.

[1] Undergraduate Student, Henry Samueli School of Engineering and Applied Sciences, Univ. of California – Los Angeles, Los Angeles, CA, 90095.
[2] Research Scientist, Aviation Systems Division, NASA Ames Research Center, Mountain View, CA, 94035.

Recent advancements in neural networks (NN) and natural language processing (NLP) offer new opportunities to automate and optimize NOTAMs to the benefit of all flight operations stakeholders. More specifically, the recent breakthrough of transformer architectures in machine learning has made huge strides in generalizing models to a variety of use cases. Transformer architectures, which are based solely on attention mechanisms, have been successfully applied to natural language processing in both large and limited datasets [4]. These breakthroughs offer huge opportunities for application in the field of NOTAMs.

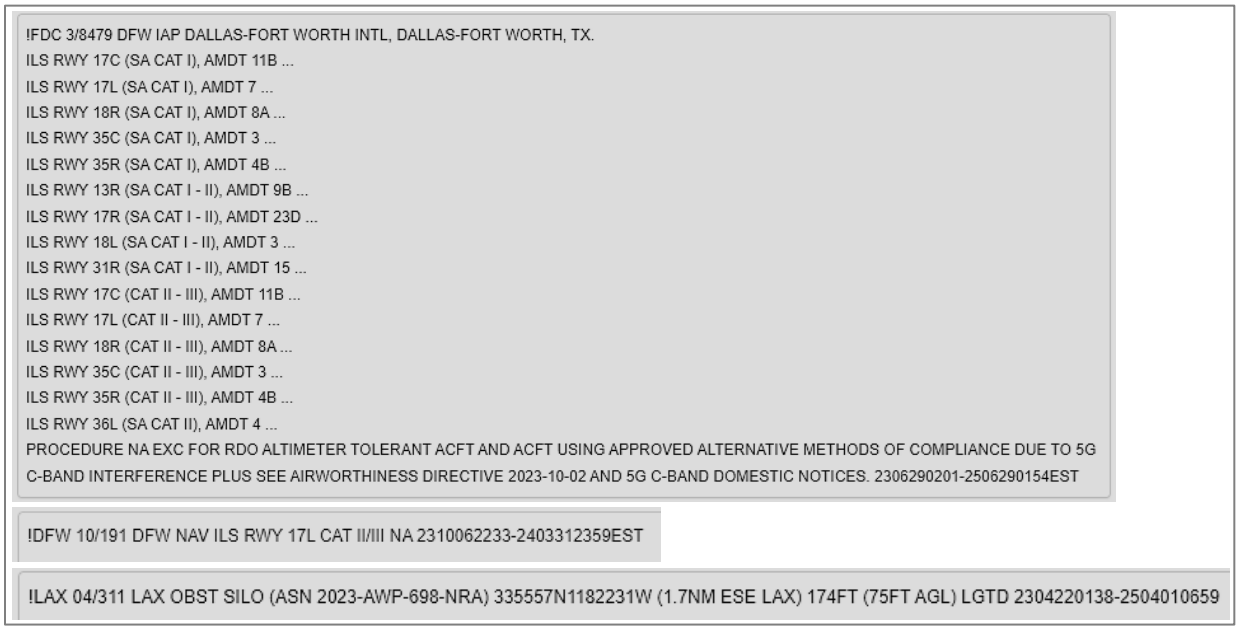


Fig. 1 Examples of NOTAMs

The feasibility study presented in this paper explores the utilization of NLP to contextualize NOTAMs into different functional categories. The motivation is derived from the potential for not only improved decision-making on part of human pilots but also seamless integration of smart unmanned aerial systems (UASs) into the national airspace (NAS). NLP can augment NOTAM interpretation on both manned and unmanned aerial systems, which can help standardize communications across the NAS. A wide range of industry applications in recent times have brought forth significant progress in NLP along with large language modeling (LLM). With the advent of newer model architectures that are substantially faster and more accurate than their predecessors, NLP has risen to the level of finding its space in mission critical applications. Effective in interpreting highly specialized expressions such as NOTAMs, as demonstrated later in the paper, NLP, and artificial intelligence (AI) in general, in aviation operations can be seen much closer than the technology of the distant future.

This paper is organized as follows: section II provides a general overview of the different ongoing efforts in the research domain for analyzing NOTAMs using non-conventional approaches. Section III presents the preliminary evaluation of multiple popular NLP architectures using different types of data. In section IV, the utilization of the best performing NLP architecture for NOTAM classification is presented. Section V summarizes the NOTAM classification test. Finally, section VI concludes the paper with a summary of findings and future directions.

II. Background

Various contemporary research exploring AI-based contextualization for complex language expressions, such as NOTAMs, can be found in the scientific community. [5] discusses the “processing and integration of time-sensitive NOTAM information over data link as well as the graphical presentation to the pilot on an EFB application” with the intention to help pilots better interpret NOTAMs. [6] discusses an approach for NOTAM processing with Natural Language Processing. It uses an attention layer on top of a traditional bi-directional RNN. Storage and graphical display of NOTAMs to improve the visibility of message information is discussed in [7]. [8] sheds light on selecting NOTAMs based on subject and status codes that are used with the selected phase of flight to determine a relevance code each NOTAM according to a set of relevance rules, which are used to help pilots filter out irrelevant NOTAMs. Self-supervised learning using BERT, leading to a structured language called Airlang, has been presented in [9]. A

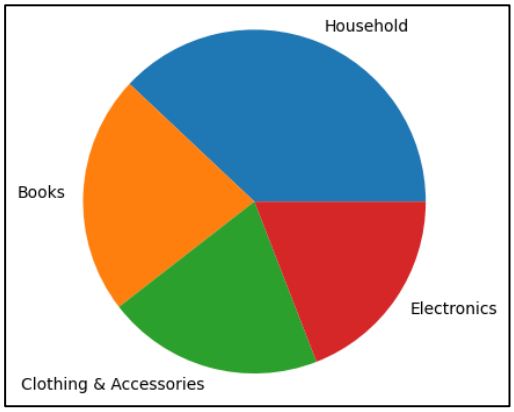
good exploration of different pre-build models such as BERT, RoBERTa, and XLNet on NOTAM data is given in [10]. Several other work on automated NOTAM interpretation, as found in [11, 12, 13, 14, 15, 16, 17], are noteworthy as they demonstrate keen interest within the aviation community for such studies and the emergence of novel approaches. In general, much of the state-of-the-art research in the NOTAM interpretation focuses on filtering out irrelevant ones prior to reception by a pilot. To achieve this, NLP methods are being explored. Furthermore, visualization of NOTAMs for the pilot upon reception using Electronic Flight Bags (EFB) is another area of interest.

Study presented in this paper aims to deliver a data contextualization framework targeting NOTAMs. The long-term vision is to have this framework generalized to more aeronautical communications methods beyond NOTAMs, such as pilot-controller conversations, ATC meeting notes, Standard Operating Procedure (SOP) references, etc. While several other feasibility studies report that NLP can be computationally impractical for such real-time needs, our approach strives to minimize the computational latency using massively parallel processing on Graphics Processing Units (GPUs).

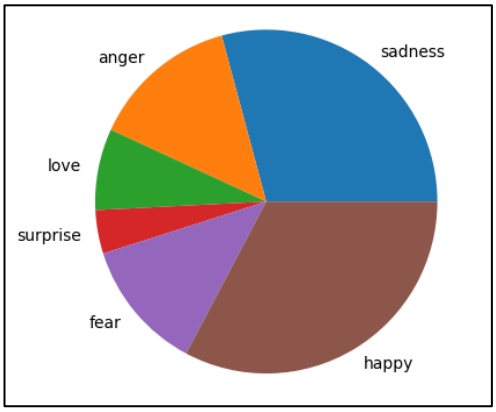
III. Feasibility Study with AI

Prior to examining NOTAM data, three unique datasets were identified with the goal of exploring the feasibility of text classification with Natural Language Processing on texts of various lengths, vocabularies, and grammars. The datasets are preprocessed individually, but each follow an 80-20 train-test split and use the same hyperparameters.

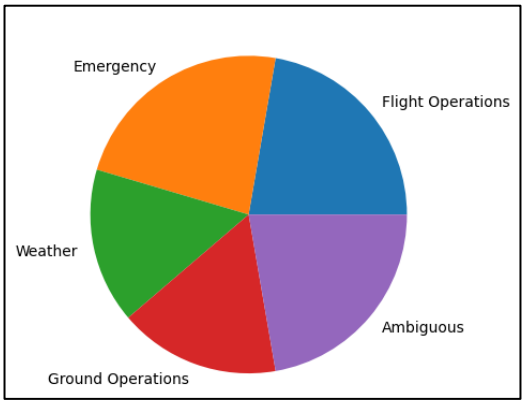
The first dataset, dubbed the *ecommerce dataset*, contains 50,425 ecommerce product descriptions (27,802 after dropping duplicates and missing values) [18]. Each instance belongs to one of four categories - Electronics, Household, Books, and Clothing & Accessories. The second collection of texts is dubbed the tweet dataset [19]. This dataset contains 21,459 short texts (21,456 after dropping duplicates and missing values) categorized into one of six labels – anger, fear, happy, love, sadness, and surprise. The third dataset, dubbed the *DRF services dataset*, contains 449 service descriptions categorized into one of five labels – Ambiguous, Emergency, Flight Operations, Ground Operations, and Weather. DRF, or Data and Reasoning Fabric, services are decentralized tools served to air mobility clients through the DRF core [20]. Fig. 2 displays the distribution of data across classes for each of the datasets.



(a) E-commerce dataset class distributions



(b) Tweet dataset class distributions



(c) DRF service dataset class distributions

Fig. 2 Different evaluation datasets and the distribution of classes in them

We perform pre-processing on all three datasets. For all three datasets we convert all characters to lowercase, strip punctuation and non-alphanumeric characters, and remove stopwords (provided from NLTK). For the *tweet dataset* we remove all "@" mentions, hashtags, retweets, links, and Unicode characters in addition to the previously applied pre-processing steps. This is specific to tweets as they are more bloated with characters that may be discarded. Lastly, each data point is padded.

Each of the first three datasets - *ecommerce dataset*, *tweet dataset*, and *DRF services dataset* are trained on four neural network architectures. The first architecture is a RNN with three layers: the first layer is a GloVe word embedding layer that uses a pre-trained word vector with 6B tokens, 400K vocab, and 300d vectors, the second layer is a LSTM with 32 units, 20% dropout, and 20% recurrent dropout, and the third layer is a dense layer with a softmax activation [21]. It uses the Adam optimizer with a learning rate of 0.001 and no weight decay. It measures categorical cross-entropy loss and tracks accuracy as its primary metric. The second architecture is again a RNN with three layers, differing from the first architecture only by the first layer: a Word2Vec word embedding layer with 300-dimensional pre-trained vectors for 3 million words and phrases [22]. Again, it uses the Adam optimizer and a learning rate of 0.001, but has a weight decay of 0.001 as well to prevent overfitting. Like the first architecture, it also measures categorical cross-entropy loss and tracks accuracy as its primary metric. The third architecture follows the same pattern; it is a RNN with three layers differing from the first two architectures only by the first layer: a pre-trained corpus called *lee_background* which is packaged with the genism FastText module [23]. This architecture uses the Adam optimizer with a higher learning rate of 0.01 and no weight decay. Like the previous two architectures, the third architecture uses categorical cross-entropy loss and tracks accuracy as its primary metric. The fourth and last architecture used is a pre-trained BERT model - *bert-base-uncased* from Hugging Face [24]. This model is pre-trained on the large corpus of English data using a masked language modeling (MLM) objective and next-sentence prediction (NSP). In all four models, each dataset is trained with a batch size of 10 and 50 epochs.

There are some hyperparameters that are not shared between models as well. Each architecture model uses early stopping with variable patience, depending on the speed of convergence. For example, the first architecture uses patience 5 for the *ecommerce dataset* and patience 3 for the *tweet dataset* and *DRF services dataset*. Additionally, the three RNN architectures use different dimensions for their embedding layers. The GloVe model uses embedding dimension 50, Word2Vec uses embedding dimension 300, and FastText uses embedding dimension 300.

The three datasets were evaluated on the 4 models based on test accuracy and training time. More specifically, test accuracy was measured as the percentage of correct model predictions in a testing dataset, and training time was measured as the number of epochs, or iterations of training, taken to train the model. Test accuracy was validated by running model inference on the testing dataset and comparing predictions to ground-truth labels. Training time was validated using training metrics automatically generated by the model.

For the *ecommerce dataset*, the BERT model had the best performance, followed closely by Word2Vec and GloVe. At convergence BERT achieved test accuracy 0.9599, Word2Vec achieved 0.9509, and GloVe achieved 0.9383. FastText lagged behind the other three models with a test accuracy of 0.8403. BERT converged the fastest, needing only 9 epochs while Word2Vec, GloVe, and FastText needed 14, 17, and 45 epochs respectively, see Fig. 3. With exclusion to BERT, which was used as a baseline measurement for the other three RNNs, Word2Vec and GloVe likely outperformed FastText because they used larger word embeddings. There were few complex words found in the product descriptions, which prevented FastText from taking advantage of its morphology based on out-of-vocabulary (OOV) vectorizations.

For the *tweet dataset*, the BERT model had the best performance, followed closely by Word2Vec and GloVe. At convergence BERT achieved test accuracy 0.9319, Word2Vec achieved 0.9212, and GloVe achieved 0.8858. FastText lacked behind the other three models with a test accuracy of 0.4419. BERT converged the fastest, needing only 6 epochs while Word2Vec, GloVe, and FastText needed 12, 42, and 50 epochs respectively, see Fig. 4. Word2Vec significantly outperformed the other two experimental architectures. The reason for FastText's poor performance likely has to do with the brevity of words in *tweets*. Many of these words consist of single char-ngrams, so FastText is unable to concatenate multiple char-ngrams to evaluate larger vectors. Since it was trained with a relatively smaller corpus, OOV words would likely skew training. The reason for GloVe's poor performance may stem from the lack of context and inter-word relationships. Again, as *tweets* are short and often include incomplete sentences, GloVe is unable to leverage word context very well.

Again, BERT had the best performance for the *DRF services dataset* followed by GloVe and Word2Vec. BERT had a test accuracy of 0.7778, GloVe had 0.7640, Word2Vec had 0.7416, and FastText had 0.3371. BERT converged in just 5 epochs, Word2Vec in 9, GloVe in 14, and FastText in 20, see Fig. 5. Because this dataset was small, there is a significant drop in training and test accuracies for all models, as overfitting was slightly present, even with an early stopping callback. Like the reasoning in the previous two datasets, FastText performed poorly because it was unable to leverage the ability to vectorize OOV words with stronger technical jargon.

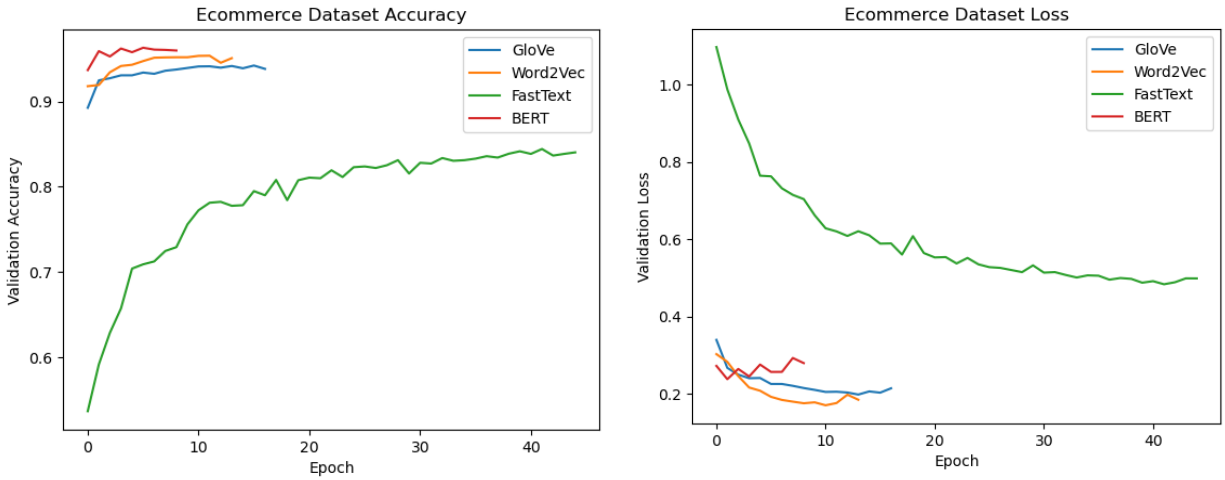


Fig. 3 Test accuracy and loss vs epochs per model for *ecommerce dataset*

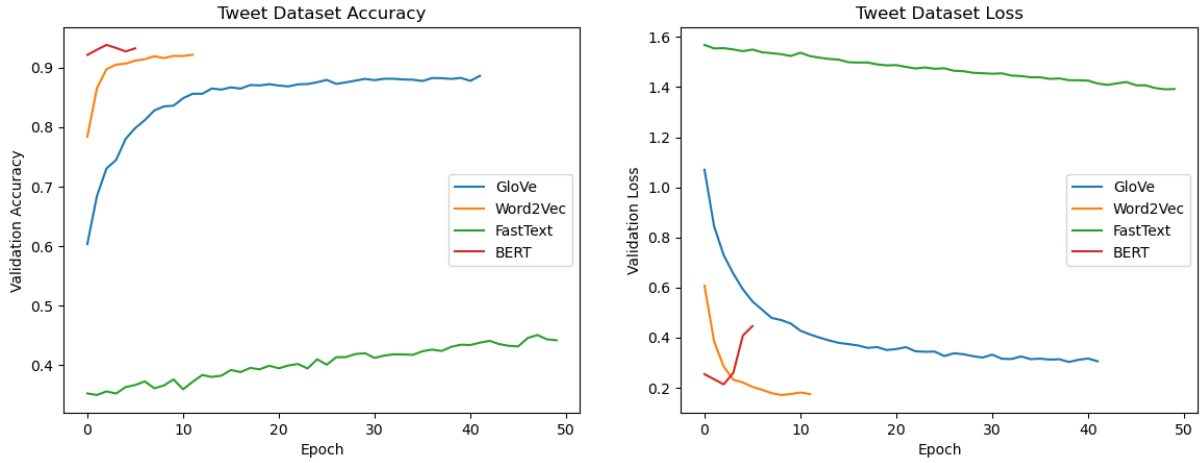


Fig. 4 Test accuracy and loss vs epochs per model for *tweet dataset*

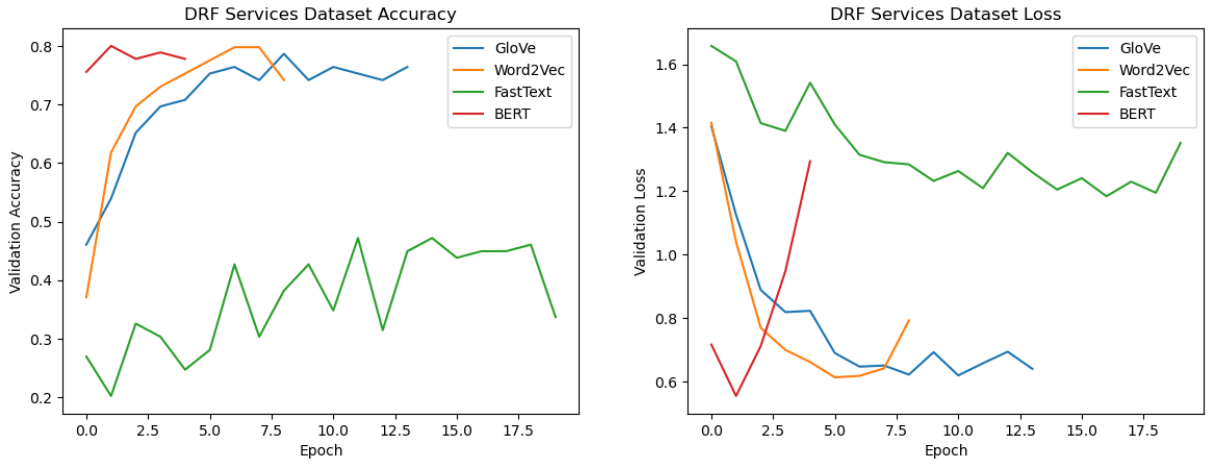


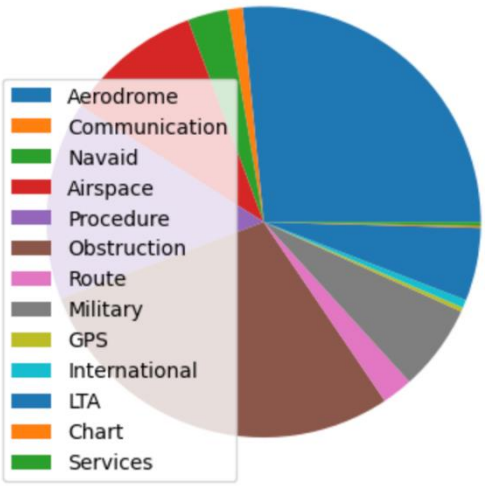
Fig. 5 Test accuracy and loss vs epochs per model for *DRF services dataset*

Based on the feasibility study of four popular models across three diverse datasets, we conclude that Natural Language Processing can be used for classifying NOTAMs. At least two of the models scored a test accuracy of 0.7500 or higher on each of the datasets, with two of the datasets resulting in accuracies well over 0.9000. With BERT performing well consistently across each dataset, we can confidently move forward with classifying NOTAMs using the same approach.

IV. NOTAM Data

Now that we know it is feasible to classify NOTAMs using Natural Language Processing, we can build a case study around real NOTAM data. We gathered open-source NOTAMs from the top 10 United States airports with the highest traffic in terms of total passenger traffic in 2021 [25]. The traffic volume at the selected airports ranges from 75.5 million total passengers at the Hartsfield-Jackson Atlanta International Airport to 37.3 million total passengers at the Miami International Airport. The active NOTAMs from each individual airport were retrieved from the official FAA website [26] using the 100 nautical mile location filter and exported to their respective CSV files. The CSVs from each airport query were then merged into one dataset. The data collected includes all 9,810 NOTAMs that have been issued and not yet expired within 100 nautical miles of the top 10 United States airports with the most total passenger traffic in 2021. We call this the *NOTAM dataset*.

Each NOTAM contains one of 13 class labels: Aerodrome, Services, Chart, LTA, International, GPS, Military, Route, Obstruction, Procedure, Airspace, Communication, or Navaid. These class labels represent the type of message being sent. They are not the same as the type of NOTAM, which is encoded in the message itself. Rather, the class labels represent the FAA-designated NOTAM series, which acts as a replacement for the domestic NOTAM subject [27]. The dataset is imbalanced, see Fig. 6, as the Obstruction, Procedure, and Aerodrome labels dominate nearly 75% of all NOTAMs while Services, Chart, International, and GPS account for a very small portion of the *NOTAM dataset*'s class labels.



Series	Name	Domestic NOTAM Subject
B	Aerodrome Movement Areas	RWY, TWY
C	Published Services	COM, WX, ATC
D	Special Activity Airspace	SAA
E	Airspace Events and Activities	PJE, Gliders, Etc.
G	Airways and Air Traffic Routes	
H	Regulatory NOTAMs	FDC, TFR, Security,
I	Apron/Ramp and Facilities	APN
J	Obstructions	OBST, Crane, BLDG, Non-FCC Tower
K	FCC Obstructions	ASR assigned
N	Ground-Based Navigational Aids	NAVAID
R	Field Condition NOTAM	RWY, TWY, APN
V	Published Instrument Procedures	IFP
Z	Satellite Based Information	GPS

Fig. 6 (a) Distribution of the NOTAM dataset, (b) FAA mapping of NOTAM series to domestic subject

The *NOTAM dataset* contained entries with an average length of 186.4 characters, with a minimum length of 14 characters and maximum length of 13468 characters. Prior to training, the dataset is filtered and normalized. All data points are flattened into a single line, and invalid NOTAMs are filtered out. We define a valid NOTAM as one that fulfills the following criteria:

1) Begins with an exclamation point followed by a valid location designator (i.e., LAX).
2) Contains a valid NOTAM number in the format MM/####.

In addition to the above criteria, we remove all NOTAMs that are categorized *GPS* or *Communication*, as the dataset only contains 33 NOTAMs labeled *GPS* and 107 NOTAMs labeled *Communication*. Both categories caused significant imbalance in the dataset and are chosen to be ignored because they make up a small portion of NOTAMs. Lastly, we identified pairs between a few classes: *Navaid* and *Route*, *Obstruction* and *Service*, and *Procedure* and

Chart. NOTAMs in each of these pairs of categories contain similar content; for example, both *Navaid* and *Route* contain navigation information. Before continuing with training, we also encoded class labels into integers. 0 mapped to *Aerodrome*, 1 mapped to *Airspace*, 2 mapped to *Navaid*, 3 mapped to *Obstruction*, and 4 mapped to *Procedure.*

After dropping duplicates and filtering out invalid NOTAMs, the *NOTAM dataset* contained 7524 data points. The next step is to pre-process the data. Each data point is flattened, replacing all newline characters with a space character. Because we want to preserve various features of the NOTAM text itself, our procedure primarily relies on the pre-trained model to handle necessary pre-processing. For example, padding is not applied in our native pre-processing step, but rather it is left to the pre-trained BERT model to handle. Moreover, we are using an uncased pre-trained BERT model which uses the input words in lowercase.

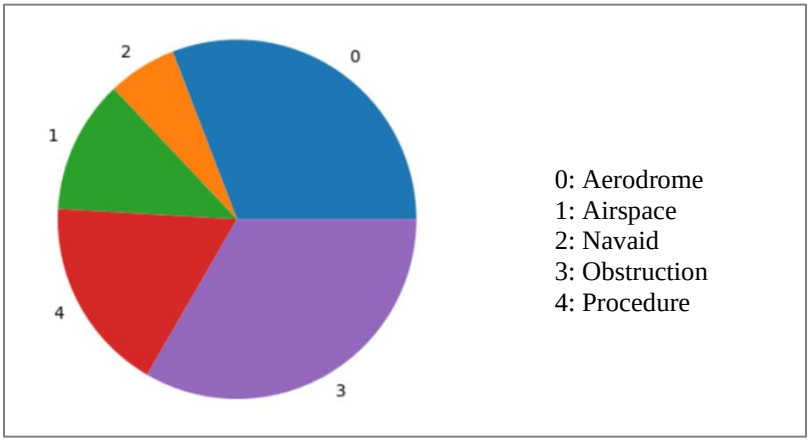


Fig. 7 Encoded label distribution after filtering

As seen in Fig. 7, the filtered dataset is more balanced than the original dataset. The 7524 NOTAMs were then split into training, validation, and testing datasets. 80% of the data was designated training, meaning that it would be used for the BERT model training. 10% of the data was designated validation, meaning that it would be used for measuring validation accuracies and loss throughout the model training. Lastly, 10% of the data was designated for testing, meaning that it would be used for model inference at the conclusion of training. The split of the data is completely random and stratified. This means that the training, validation, and testing datasets contain representation from each class label, while the contents of the datasets themselves are randomly distributed.

V. NOTAM Classification

The model chosen for NOTAM classification is the pre-trained BERT model - *bert-base-uncased* from Hugging Face [24]. As mentioned in the feasibility study, this model is pre-trained on the large corpus of English data using a masked language modeling (MLM) objective and next sentence prediction (NSP). Masked language modeling randomly masks 15% of the words from the model training and reserves them for predictions, allowing the model to learn a bidirectional representation of a sentence. With next-sentence prediction, two masked sentences are concatenated, and the model is trained to predict whether the sentences follow each other immediately. Ultimately, BERT learns the inner representations of a corpus.

Our approach for classifying NOTAM data uses transfer learning. The BERT model is pre-trained on a dataset consisting of 11,038 unpublished books and English Wikipedia [28] and consists of 110M parameters. This is helpful, as the model learns an inner representation of the English words that exist within a NOTAM. We fine-tune the model using the *NOTAM dataset* without freezing meaning that everything within the BERT model is updated during training, including the token embeddings and encoders. With this fine tuning, the model can extract NOTAM-specific jargon in addition to English words present in NOTAM messages.

We keep many of the default hyperparameters used in the initial training of the BERT model. The model was trained for 1 million steps with batch size 256 while limiting sequence length to 128 tokens for 90% of the steps and 512 tokens for the remaining 10%. Each token is a sequence of characters grouped together as a semantic unit for processing. It uses the Adam optimizer with an initial learning rate of 1e-4, a weight decay of 0.01, and a learning rate warmup of 10,000 steps. After the warmup, learning rate is linearly decayed. For fine-tuning on the *NOTAM dataset*, this pre-trained BERT model uses subset accuracy score as its computation metric, which is equivalent to *the number of predicted labels matching ground truth / the number of total samples*. We also considered using the F1 score, which

is equivalent to *2 * (precision * recall) / (precision + recall)*. In an unbalanced dataset, pure accuracy score is misleading as labels with few samples may be classified incorrectly but will not substantially decrease the accuracy score. F1 score performs better on unbalanced datasets because it measures performance in both the precision and recall of a model. In the case of the *NOTAM dataset*, the dataset was slightly unbalanced. However, performance when training with accuracy score and F1 score did not make a significant difference, as testing accuracy reached higher than 98% in both computation metrics. We also use an early stopping callback of patience 1 for the model. This means that training will conclude when an epoch reaches an accuracy that is less than or equal to an accuracy it has previously reached. We noticed that the BERT model converged very quickly, reaching an accuracy of *0.986720* after the first epoch. Because of the fast convergence, early stopping with patience 1 was a reasonable choice. The purpose of using an early stopping callback is to prevent overfitting once convergence is reached. We fine-tuned the model with per-device batch size 10 and 50 epochs (assuming it did not stop early due to the early patience callback). The evaluation strategy we use is epochs.

Table 1: NOTAM model training progress summary

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.060482	0.986720
2	0.121300	0.045256	0.992032
3	0.121300	0.034074	0.993360
4	0.028900	0.040537	0.994688
5	0.019500	0.047369	0.994688

The model was trained using a Dual NVIDIA RTX6000 Ada Generation Graphics Card with 48GB of GDDR6 memory. It was trained over NVLink using Cuda 11.8. With this hardware, the total training time only took 8.35 minutes with an average epoch training time of 1.25 minute. As seen in the chart above, the model reached an accuracy score of 0.986720 after the first epoch and finished after five epochs with an accuracy score of 0.994688. There was no improvement in accuracy score between epochs four and five, and the model terminated due to early stopping. After the model was trained, we measured accuracy score on the testing dataset. The testing dataset was a stratified, random 10% split of the original *NOTAM dataset* that the model was not trained on. The resulting testing accuracy was 0.996, meaning that 99.6% of the samples in the testing dataset were classified correctly.

We created an inference module to utilize the trained model. It provides a user-friendly method of utilizing the trained model. The inference module is a command-line Python tool that takes in NOTAMs as input from the user and produces a class prediction using the model. The tool also presents the confidence level of the prediction. The module can consistently produce predictions in ~0.1 seconds, enabling real-time classification in future applications.

Fig. 8 shows an example of the inference module at work. It predicts each of the three user-inputted NOTAMs correctly with a confidence of 99% or greater for each one. The three NOTAMs are normalized and classified in 0.4 seconds, 0.07 seconds, and 0.06 seconds. Detections are highlighted via yellow outlines in Fig. 8. The higher detection time for the first inference example stems from the one-time model initialization overhead that is taken when the application starts.

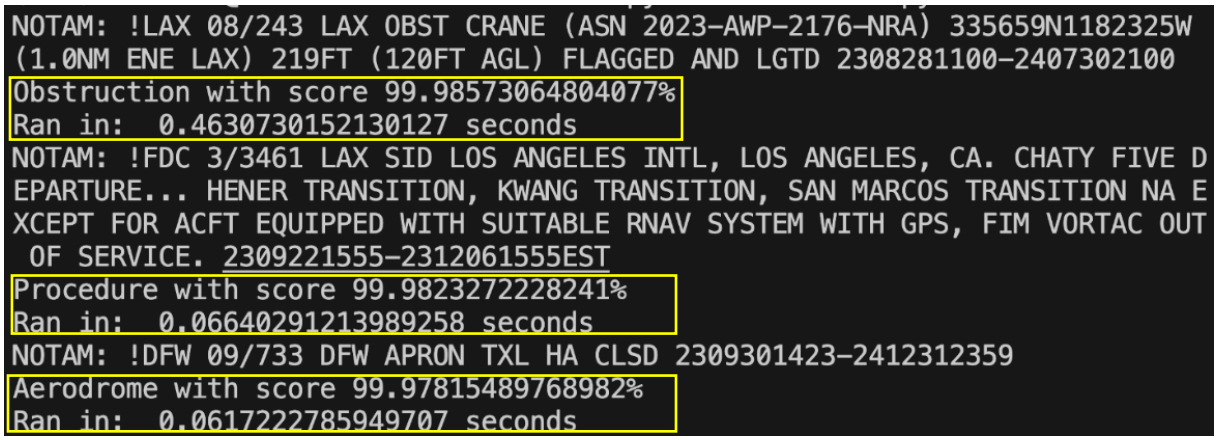


Fig. 8 Inference module example

VI. Conclusion and Future Work

The feasibility study to classify NOTAMs using NLP-based approach offers promising results, as we see close to 99% accuracy in classification. Future efforts in this research will be dedicated to expanding the dataset to include a wider NOTAM pool from different regions across the US as well as international air operation sectors. Additionally, the NOTAM classifier will be made available as a service on the NASA DRF platform for airspace stakeholders to subscribe to and avail as an inline interpreter of NOTAMs for human and machine consumption. Given the versatility of the NLP-based classification approach in interpreting unstructured text, the utility of this research outcome can be extended to other application areas such as categorization of aviation service descriptions on open data marketplace, standard operating procedures (SOPs), and pilot-controller conversation transcripts.

References

[1] "What is a NOTAM?," FAA, 2021. [Online]. Available: https://www.faa.gov/about/initiatives/notam/what_is_a_notam. [Accessed 2023].

[2] "NOTAM," Wikipedia, 2023. [Online]. Available: https://en.wikipedia.org/wiki/NOTAM. [Accessed 2023].

[3] "NOTAMs," FAA, 2023. [Online]. Available: https://www.faa.gov/air_traffic/flight_info/aeronav/notams/. [Accessed 2023].

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need," *Neural Information Processing*, 2017.

[5] A. Sindlinger, N. Zimmer, T. Wieseman, R. Li, M. Andersson and S. van der Stricht, "Automated NOTAM processing for a graphical and textual integration on data link equipped aircraft," in *Integrated Communications, Navigation and Surveillance Conference, ICNS*, Herndon, 2010.

[6] B. Mi, Y. Fan and Y. Sun, "NOTAM Text Analysis and Classification Based on Attention Mechanism," in *Journal of Physics: Conference Series*, 2022.

[7] T. Yasunaga and 貴紀 安永, "Notam message processing system". Japan Patent JPH0896151A, 10 February 1998.

[8] N. Zimmer and K. Bayram, "Selective NOTAM Notification". United States Patent US20150170492A1, 21 November 2017.

[9] A. Arnold, F. Ernez, C. Kobus and M.-C. Martin, "Knowledge extraction from aeronautical messages (NOTAMs) with self-supervised language models for aircraft pilots," in *North American Chapter of the Association for Computational Linguistics*, Seattle, 2022.

[10] S. S. B. Clarke, P. Maynard, J. A. Almache, S. G. Kumar, . R. Rajkumar, A. C. Kemp and R. Pai, "Natural Language Processing Analysis of Notices To Airmen for Air Traffic Management Optimization," in *AIAA Aviation Forum*, 2021.

[11] BBN Technologies, "INTELLIGENT SEMANTIC QUERY OF NOTICES TO AIRMEN (NOTAMs)," AFRL, 2006.

[12] K. K. Patel, G. Desaulniers, A. Lodi and F. Lecue, "Explainable prediction of Qcodes for NOTAMs using column generation," 2023.

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

[14] E. T. Evans Jr, S. D. Young, T. S. Daniels and R. R. Myer, "Usability of EFBs for Viewing NOTAMs and AIS/MET Data Link Messages," in *Conference on Digital Avionics Systems (DASC)*, 2013.

[15] D. Steiner, F. Burgstaller, E. Gringinger, M. Schrefl and I. Kovacic, "IN-FLIGHT PROVISIONING AND DISTRIBUTION OF ATM INFORMATION," in *Congress of the International Council of the Aeronautical Sciences*, Daejeon, 2016.

[16] L. Yinhui and D. Yipeng, "Recognition and Processing of NATOM Based on Decoupling Features and Classifiers," 2021.

[17] C. Yang and C. Huang, "Natural Language Processing (NLP) in Aviation Safety: Systematic Review of Research and Outlook into the Future," *Aerospace,* 2023.

[18] S. Shahane, "Ecommerce Text Classification," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/saurabhshahane/ecommerce-text-classification. [Accessed 2023].

[19] I. Juyal, "Emotions in text," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/ishantjuyal/emotions-in-text. [Accessed 2023].

[20] M. Abdelbaky, J. Chen, A. Fedin, K. Freeman, M. Gurram, A. K. Ishihara, C. Joe-Wong, C. Knight, K. Krishnakumar, C. Robins, C. Robinson, P. Shannon, S. D. Shetye and L. Tomljenovic, "DRF: A Software Architecture for a Data Marketplace to Support Advanced Air Mobility," in *AIAA Aviation 2021 Forum*, 2021.

[21] J. Pennington, R. Socher and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, 2014.

[22] Google, "word2vec," Google, 2013. [Online]. Available: https://code.google.com/archive/p/word2vec/. [Accessed 2023].

[23] Gensim, "FastText Mode," Gensim, [Online]. Available: https://radimrehurek.com/gensim/auto_examples/tutorials/run_fasttext.html. [Accessed 2023].

[24] Hugging Face, "bert-base-uncased," Hugging Face, [Online]. Available: https://huggingface.co/bert-base-uncased. [Accessed 2023].

[25] WorldAtlas, "The 10 Busiest Airports In The United States," WorldAtlas, 2021. [Online]. Available: https://www.worldatlas.com/places/the-10-busiest-airports-in-the-united-states.html. [Accessed 2023].

[26] "FNS NOTAM Search," FAA, [Online]. Available: https://notams.aim.faa.gov/notamSearch/nsapp.html#/. [Accessed 2023].

[27] Federal Aviation Administration, "ICAO NOTAM 101 for Airport Operators," Federal Aviation Administration, 2021.

[28] Wikipedia, "English Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/English_Wikipedia. [Accessed 2023].