DeBERTa-AT: A DeBERTaV3 Variant Fine-Tuned on Air Traffic Data

1st Richard Yue
Northeastern University
NASA Ames Research Center
Moffett Field, USA
richard.s.yue@nasa.gov

2nd David Nielsen
KBR, Inc.
NASA Ames Research Center
Moffett Field, USA
david.l.nielsen@nasa.gov

3rd Krishna M. Kalyanam
Aviation Systems Division
NASA Ames Research Center
Moffett Field, USA
krishna.m.kalyanam@nasa.gov

***Abstract*—Large language models (LLMs) offer a powerful platform and can leverage tools to extract relevant information and provide recommendations for air traffic users. These can range from classification of voluntary safety reports to discovering shared successful corrective actions and creating more accurate transcriptions of air traffic control conversations. However, using LLMs requires leveraging the knowledge contained in legacy aviation datasets, which can be time-consuming and compute-intensive. This paper describes the creation of DeBERTa-AT, a variant of the DeBERTaV3 model fine-tuned on air traffic data. DeBERTa-AT continues pretraining using the replaced token detection (RTD) pretraining task with gradient-disentangled embedding sharing (GDES) to offer improved performance and faster training convergence on downstream tasks in the aviation field.**

DeBERTa-AT is evaluated on two downstream binary classification tasks: aviation-specific constraint classification and document classification, with text datasets created from letters of agreement (LOAs). During downstream fine-tuning, improvements were shown in performance as compared with the DeBERTaV3-base model with frozen embeddings. The results show that models combined with sample-efficient continual pretraining can offer substantially better results on domain specific data when less training data is available.

***Index Terms*—DeBERTaV3, Replaced Token Detection, Pretraining Techniques, Random Initialization**

I. Introduction

Due to the large amount of natural language data used in aviation, there has been growth in the application of large language models (LLMs). Aviation industry stakeholders, however, have specialized vocabulary, and there is strong demand for high quality LLMs in the aviation domain, as seen in section II. Additionally, regulatory constraints often prevent aviation industry data from being made publicly available. Training models such as those in the BERT family, of which DeBERTa is a member, can require major compute. This work aims to train a highly effective foundation model for the aviation domain, which could be used for faster and more sample-efficient future fine-tuning or inference.

As shown with ELECTRA [3] and DeBERTa [1], using a generative and discriminative model in tandem during pretraining can greatly improve contextual representations of tokens and can lead to improved performance on downstream tasks as compared to training on masked language modeling (MLM) alone. This process is known as replaced token detection (RTD). Furthermore, DeBERTaV3 [2] showed that disentangling embedding sharing so that gradient updates flow only from the generator to the discriminator model further improves downstream performance and convergence speed. The improved representations obtained using these pretraining techniques can offer substantial gains in performance and reduce the compute and time needed to train further downstream as new data is collected or new tasks arise. This work demonstrates how to adapt RTD pretraining to a domain-specific application. It could easily be used to train foundation models in other technical domains as well.

Although ELECTRA and the different versions of DeBERTa offer good performance on tasks like GLUE [10], they are trained on general domains and do not include specialized vocabulary present in the aviation domain. This work makes the following contributions:

1) It demonstrates how to train DeBERTa-AT, a new model that displays performance gains on aviation-specific tasks when compared to DeBERTaV3-base. DeBERTa-AT is a model fine-tuned using the ELECTRA pretraining objective, also known as replaced token detection (RTD) and described in Section III, on a corpus of aviation-specific text.

2) Extensive experiments validate the performance of the fine-tuned model versus the base DeBERTaV3 model with frozen embeddings to show the effect of RTD continual pretraining on aviation-specific vocabulary.

3) The performance of DeBERTa-AT is evaluated on two aviation-specific classification tasks, and strong results are achieved in both cases.

II. Related Work

The Aviation Safety Reporting System (ASRS), a voluntary anonymous safety incident reporting system, was analyzed

using summarization, sentiment analysis and clustering to find common corrective actions [6]. This used both generative models and those from the Bidirectional Encoder Representations from Transformers (BERT) family of models [4]. These have also been used to format aviation domain text to make it more correct and interpretable by aviation domain experts [8]. Work has been done to make aviation-specific BERT-variants such as Aviation-BERT, which served as motivation for this paper [7]. Although these techniques worked well, training using standard BERT techniques is not as effective when training data is scarce. To address this issue, we employ the more sample-efficient RTD technique described in Section III.

Clark et al. [3] demonstrate how a generator and discriminator model can be used in tandem to further refine masked language model representations. ELECTRA paved the way for new, sample-efficient pretraining techniques. He et al. [2] present DeBERTaV3, which uses gradient-disentangled embedding sharing (GDES) and the RTD pretraining objective described in Section III to further improve upon the original DeBERTa paper. However, these models were trained on general corpora such as Common Crawl[1], Wikipedia[2], and the Book Corpus [9]. DeBERTa-AT uses the RTD pretraining objective to improve domain-specific representations and is trained on highly specialized aviation data. Like DeBERTaV3, it uses GDES during training so that the generator shares its refined embeddings but not the other way around, offering greater performance and faster convergence.

Beltagy et al. [11] present SciBERT, a model pretrained using the same configuration as BERT on highly technical scientific literature. This paved the way for domain-specific pretrained models that could be fine-tuned more easily and effectively downstream for uses in similar domains. While DeBERTa-AT employs a similar technique of fine-tuning a pretrained model on domain-specific data, SciBERT employed traditional masked language modeling, an older technique. DeBERTa-AT instead uses the more effective and sample-efficient RTD training method, which works when data is scarce and offers a starting point for future fine-tuning in the aviation domain.

Wei et al. [13] show that the way random initialization occurs can have a major impact on convergence. The most common methods mentioned are to randomly initialize weights from a normal or a uniform distribution. Depending on use case, one distribution can lead to better performance when converging. Simply changing the distribution type was shown to be ineffective in the case of DeBERTa-AT, which displays a high degree of sensitivity to randomization. Instead, its fine-tuning process employs a large number of trials with varying random seeds to demonstrate variability due to differences in initialization parameters and to determine the path towards optimal performance, offering more coverage of the hyperparameter space.

Saxe et al. [12] investigate the effect of weight initialization and other nonlinear learning parameters on the performance of linear deep neural networks. They find that when the weight initialization matrix is orthogonal, training time is depth-independent, which shows that the right random initialization values obtained from a random seed can have a major effect on speed of convergence. While novel, this technique applies more readily to simpler models that are initialized from scratch and do not include pretrained weights. DeBERTa-AT uses pretrained embeddings and rigorous selection of a random seed showed more demonstrable gains in performance.

Narkhede et al. [14] discuss various schemes for parameter initialization and evoke the importance of such considerations on convergence speed and results. In addition to discussing both random and data-driven weight initialization, they discuss hybrid initialization approaches which blend both. These include using standard deviation or n-gram embedding values farthest from 1 when choosing the range of values for weights. They further adjust weight matrices to obtain orthonormal values, which shows promise in improving the accuracy and convergence of recurrent neural networks (RNNs). DeBERTa-AT draws inspiration from the idea of orthogonality, but it operates in the domain of pretrained models and continual pretraining, where the vast majority of weight matrices cannot be initialized from scratch. The most effective technique was shown to be combining random seed trials with frozen embeddings meant to contain a high density of encoded information (which approaches the full rank of an orthogonal matrix).

Makkuva et al. [15] assess the effect of initialization distributions on the convergence of a single layer transformer model. They prove via multiple theorems that data properties and parameter initialization play a significant role in the convergence of transformer models. The model considered is a single-layer transformer. The authors conclude that full rank initialization converges to low rank, as does starting with a rank 1 matrix; both extremes lead to local minima, and conjoining proper randomization techniques with a data-driven approach can prevent poor convergence. However, it is not possible to get the most of these techniques when weights have already been pretrained. DeBERTa-AT instead makes use of similarly inspired data-driven techniques (preserving pretrained embeddings) and randomization techniques (a set of random seed trials) to ensure conditions are optimal for effective and efficient convergence near global optima. The strong results demonstrate the effectiveness of this method.

III. Methods

Encoder Models In the context of this work, an encoder model is defined as a model that uses only the encoder portion of the traditional encoder-decoder transformer model[3]. A prime example of this is the BERT [4] family of models, all of which operate using a self-attention mechanism [16]. In other words, a model can access information about all the tokens in a sentence simultaneously when making decisions, such as classifications. Encoder models train rich contextual

[1]https://commoncrawl.org/

[2]https://www.wikipedia.org

[3]https://huggingface.co/learn/nlp-course/en/chapter1/5

representations of tokens that are passed into them. With BERT, this is traditionally achieved via masked language modeling [4], where a portion of the input tokens are removed and the model is tasked with predicting the most likely tokens to fill the gaps.

DeBERTaV3 In this work, the model selected is DeBERTaV3, from the BERT family. DeBERTaV3 is a bidirectional transformer encoder model based on the BERT architecture. It offers two enhancements to BERT: disentangled attention (DA) and an enhanced mask decoder [1]. Unlike previous models that use a single vector to represent both word content and the position of words in a sequence, DeBERTaV3 separates these into a semantic vector and a positional encoding vector that represents the relative positions of each token in the sequence. When attention scores are computed, this is done in two distinct matrices—one for the word content and one for relative word position. Like BERT, DeBERTaV3 uses masked language modeling (MLM) during pretraining. Since the disentangled attention only accounts for relative positions of tokens, absolute positions are handled by an enhanced mask decoder, which accounts for the absolute position of tokens in the decoding layer. An additional innovation in training for all DeBERTa models is the use of the RTD pretraining objective.

RTD Pretraining ELECTRA [3] is a model trained in a two-step process: In the first step, a generator model is trained on the MLM objective, with 15% of tokens randomly masked in each iteration. During this process, the generator learns to predict which tokens are likely to occur given the surrounding context. In the second step, a separate discriminator model is trained to detect which tokens have been replaced via masked language modeling. This pretraining method shows significant gains compared to models trained on MLM alone. The RTD architecture and steps are shown in Figure 1 below. DeBERTa-AT is trained using this method on aviation-specific data. Work done on ELECTRA also showed that performance further improved when the adversarial relationship between the generator and discriminator was softened and embeddings (the first three layers of each model) were shared between the two. This work replicates a specific type of embedding sharing called gradient-disentangled embedding sharing, originally detailed in the DeBERTaV3 paper.

Gradient-Disentangled Embedding Sharing (GDES) A major improvement with DeBERTaV3 was gradient-disentangled embedding sharing. In the original ELECTRA paper, updates to embeddings were shared from the generator to the discriminator and then back again in the other direction. This allows for alignment between the two models during training, but it also leads to a tug-of-war dynamic in which the generator tries to bring similar embeddings together while the discriminator tries to pull embeddings apart in order to detect words that don't belong. This pushes the embeddings in different directions and slows down training.

1) To investigate these issues, the DeBERTaV3 authors [2] compare three embedding techniques: **Embedding sharing** is identical to the original ELECTRA paper method, where embeddings are shared between both models in both directions. This presents the aforementioned difficulty of competition between the generator and the discriminator during training.
2) **No embedding sharing** is when the generator and discriminator each maintain separate embeddings during training. This prevents the tug-of-war but raises a new problem: the generator and discriminator can move in different directions, reducing the effectiveness of the final model.
3) **Gradient-disentangled embedding sharing (GDES)** GDES shares embeddings from the generator to the discriminator but not in the other direction. When the generator refines representations with MLM, the embedding layers are copied to the discriminator model so the generator has a hold on how token types are represented in the embedding space. Meanwhile, the discriminator updates other layers to coincide with new representations provided by the generator in each batch.

GDES is shown in the DeBERTaV3 paper to greatly improve both convergence speed and the performance of the final model when compared with the other two methods [2].

Corpus The training corpus for this work was built from FAA Letters of Agreement (LOAs). The documents formalize operations between airspace users across the National Airspace System (NAS), which provides an excellent source of aviation terminology to use for continual pretraining. The LOAs are agreements between two or more airspace users and cover many different types of operations in many regions. This variety provides a large amount of formal but natural language that captures the breadth of aviation terminology, which serves as the basis for the RTD pretraining task. Section IV provides the details for how this training corpus was processed for use with RTD continual pretraining.

IV. Experimental Setup

Pretraining Dataset The LOAs are stored in portable document format (PDF) and the corpus contains 7,497 LOA PDFs. These PDFs were processed into structured JavaScript Object Notation (JSON) files containing the raw text using Amazon's Textract service.[4] The raw text was gathered for each LOA document and tokenized. The DeBERTaV3 model is limited to input sequences of 512 tokens but many of the tokenized LOA documents exceeded this length. In order to utilize the entirety of the pretraining data and avoid truncation, the preprocessed documents were divided into a 500 token sliding window with an overlapping stride of 50 tokens. This chunking process resulted in 38,442 pretraining documents of the correct size to be used in the training of DeBERTaV3, as described in Section V-B. These training documents were divided using an 90%/10% split into training and validation sets for use in the RTD task. This validation set was used to track the RTD performance per epoch during the fine-tuning process.

[4]https://docs.aws.amazon.com/textract/latest/dg/what-is.html

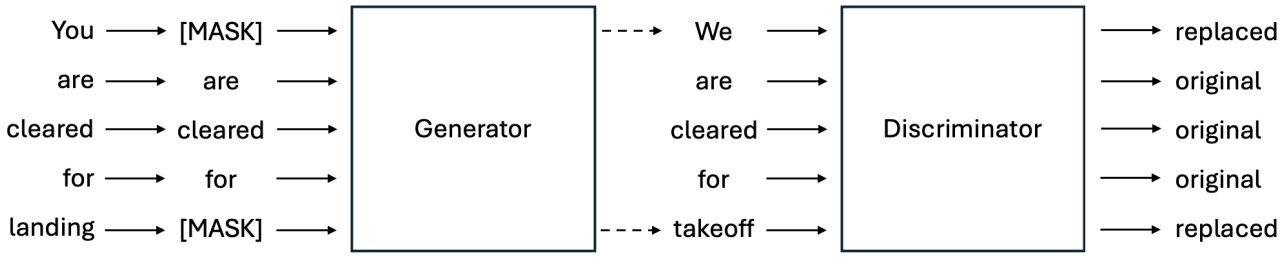

Fig. 1. RTD pretraining architecture example processing

V. Pretrained DeBERTaV3 variants

A. DeBERTaV3

The weights used are pretrained weights from the DeBERTaV3-base model[5] released by Microsoft in the original DeBERTaV3 paper [2]. The vocabulary used is the base vocab for DeBERTaV3-base and served as the basis for fine-tuning to the aviation domain using the pretraining dataset as described in the following section.

B. DeBERTa-AT

As mentioned in sections I and II, this paper describes the creation of DeBERTa-AT from the original DeBERTaV3 base model using the same RTD process [3] from the original work's pretraining objective. The HuggingFace Pytorch Trainer is used to train DeBERTaV3 on the selected corpus from the original DeBERTaV3-base configuration file, with pretrained weights loaded. As in the ELECTRA paper, a generator and discriminator model are instantiated for use in an adversarial-like setting. While the generator trains to fill in masked tokens, the discriminator is tasked with detecting which tokens were replaced and which are part of the original text. For training with a generator and discriminator, two distinct models are instantiated with the same DeBERTaV3 checkpoint, one for each task. Although the original ELECTRA work [3] indicates that a smaller generator can be used to speed up training, it was found for this use case that the best performance is achieved when the generator and discriminator are identical. The casing used is the standard casing implemented in the DeBERTaV3 default tokenizer. Training was accomplished using a single Nvidia H100 Tensor Core GPU on the NASA Advanced Supercomputing Cluster. Training the DeBERTa-AT model in this setup takes a total of 30 hours for 90 epochs. All models are saved using the HuggingFace Trainer API with corresponding bin, Safetensor, and tokenizer files. This makes them compatible with transformers library AutoModel classes. All models are implemented in PyTorch. The full fine-tuning process is described in section VI.

VI. Finetuning DeBERTaV3

Hyperparameters are selected using a shallow grid search and take inspiration from similar masked language model and sequence modeling training setups such as those found on HuggingFace[6] [7] [8]. Furthermore, as discussed in section VIII, a set of 804 trials is performed to investigate the performance and variability when starting from different random seeds.

For the generator model, an AutoModelForMaskedLanguageModeling is instantiated, as well as a data collator for masked language modeling with the mask probability set to 0.1. For the discriminator model, an AutoModelForTokenClassification is loaded with the number of labels set to 2. This enables a binary classification for each token that decides whether that token was replaced. For the classification tasks, the transformer library AutoModelForSequenceClassification model head is used with two labels, which passes linear output from the classification layer into a softmax layer.

The maximum sequence length is 500 for all but one classification task described in section VI-A. All tasks use a weight decay of 0.01, beta1 of 0.9, beta2 of 0.999 and the PyTorch implementation of AdamW as the optimizer. Training is performed for 30 epochs, with learning rates scaling up from 2e-06 to 2e-05. For each task, the validation set is used in order to determine the best performing hyperparameters and random seed. The best learning rate for training is determined to be 2e-06.

A. Tasks

Two aviation-specific tasks were selected for the chosen experiments in order to evaluate if the fine-tuning process improves results on domain tasks. These tasks were derived from the efforts to digitize LOAs and represent use cases for LLMs within the natural language technical documents [17]. If improvement is shown on these tasks, DeBERTA-AT demonstrates utility within the aviation domain. Both tasks are binary classification tasks and the labeled data is drawn from LOAs that were held out from the pretraining dataset which is then divided into a 70%/20%/10% train/validation/test split. The two tasks are:

1) Document classification
2) Constraint classification

Document classification is a two-class classification process that divides documents into 'civil' or 'not-civil' documents.

[5]https://huggingface.co/microsoft/deberta-v3-base

[6]https://huggingface.co/docs/transformers/en/tasks/masked_language_modeling

[7]https://huggingface.co/docs/transformers/en/tasks/sequence_classification

[8]https://huggingface.co/docs/transformers/en/tasks/token_classification

'Civil' documents are those whose signatories are all public entities like the FAA while 'not-civil' documents have one or more signatories who are non-public airspace users such as private companies. A set of 493 LOA documents were labeled by a subject matter expert (SME) and the counts of each class can be seen in Table I. This task came about because of the work to digitize LOAs focused on these civil LOAs [17] and training NLP models to classify documents is important for the automation of this step.

TABLE I
COUNT OF DOCUMENT CLASS LABELS

Total documents	Civil	Not-civil
493	222	271

Secondly, constraint classification is the process of identifying which portions of an LOA contain constraints on an airspace user's trajectory, e.g., a limit on an aircraft's altitude at a certain navigational point. 499 individual LOA lines from 222 LOAs were reviewed by a SME and labeled as containing a constraint or not and the resultant labels can be seen in Table II. This classification task also supports the work laid out for LOA constraint classification [17].

TABLE II
COUNT CONSTRAINT LINE LABELS

Total lines	Constraint	Not-constraint
499	129	370

VII. FROZEN DEBERTAV3 EMBEDDINGS

In a similar fashion to the SciBERT paper [11], the first three layers of the base model are frozen when training an initial classifier for each task. This allows for a baseline to be obtained from the DeBERTaV3 general domain embeddings for comparison with the fine-tuned variant. Hyperparameter settings are replicated from section VI and results are provided for the best performing learning rates.

VIII. RANDOM SEED TRIALS

In order to further explore the hyperparameter space and gauge variability to quantify the probability of improving performance on the chosen classification tasks, a total of 804 random seed trials are run, one for each seed in the range $\{0..200\}$. This experiment is repeated for each selected learning rate. Average accuracy and F1 scores are presented across all seeds and the best performing seed and its results are shown in section IX.

Sensitivity to Random Initialization Due to gradient instability, ultimate model performance is contingent upon the selection of a suitable random seed during initialization. Furthermore, due to round-off errors, even the same seed can produce different results. To mitigate this, full determinism is enabled by setting the determinism flag when setting the seed, at the expense of training speed.

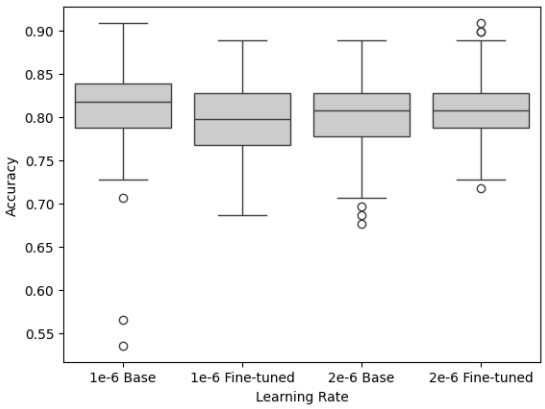

Fig. 2. Document classification accuracy comparison by learning rate

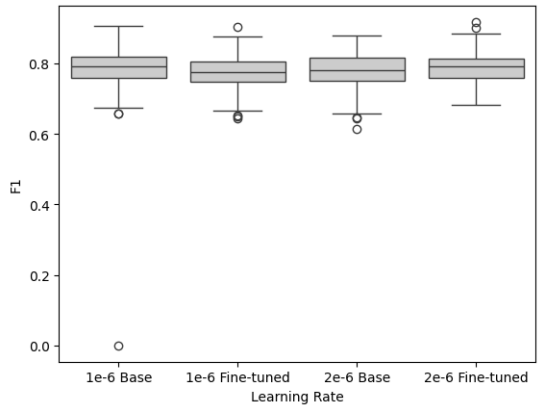

Fig. 3. Document classification F1 comparison by learning rate

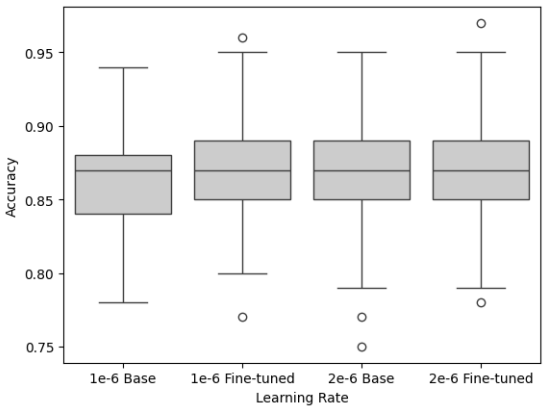

Fig. 4. Constraint classification accuracy comparison by learning rate

TABLE III
DOCUMENT CLASSIFICATION BEST SCORES

Model	Learning Rate	Accuracy	F1
DASC 2024	1e-04	0.86	0.85
Base	1e-05	0.909	0.905
Fine-tuned	2e-06	0.909	0.917

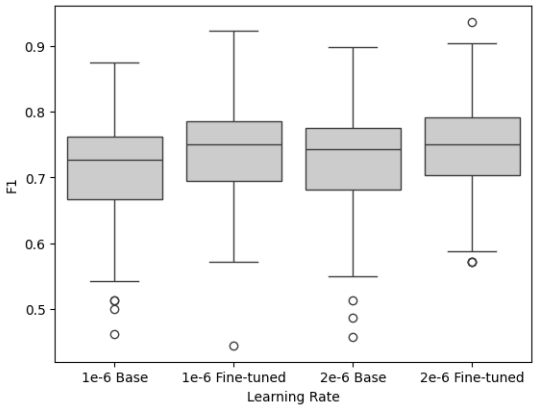

Fig. 5. Constraint classification F1 comparison by learning rate

TABLE IV
CONSTRAINT CLASSIFICATION BEST SCORES

Model	Learning Rate	Accuracy	F1
DASC 2024	1e-04	0.94	0.91
Base	1e-05	0.96	0.916
Fine-tuned	2e-06	0.97	0.936

IX. RESULTS

The performance of this model, visible in Figures 2, 3, 4, and 5, and Tables IV and III, improves upon previous work [5], which utilized less complex transformer models as seen in the tables. Document classification shows less separation between base DeBERTaV3 and the fine-tuned DeBERTa-AT than was seen in constraint classification, as illustrated by Figures 2 and 3. This is possibly due to the information needed to distinguish civil vs. non-civil documents being more general and present in a general English corpora than the distinction in the other classification task. However, there is still an increase in the F1 score for the best performing model as shown in Table I. This shows that there is information gained by the RTD process even if it is not as pronounced.

These results demonstrate how in the aviation domain, training newer, more advanced models using sample-efficient pretraining techniques can offer better results, particularly when less data is available. This work set out to train a highly effective foundation model for future downstream fine-tuning in the aviation domain and this was accomplished using the selected methods. It also outlines the process to adapt RTD pretraining to a domain specific application. This process could therefore be replicated in additional non-aviation technical domains or with additional domain specific documents.

X. CONCLUSION

This paper showed the benefits of creating DeBERTa-AT by continual pretraining of the DeBERTaV3 model using aviation data. It demonstrated the utility of DeBERTa-AT and how it can provide a better basis for aviation domain tasks than LLMs trained only on general language corpora.

Future work to further improve performance on aviation domain tasks may include experimentation with other DeBERTa variants such as DeBERTaV3-large, additional hyperparameter-tuned versions, a distilled version, or additional task-specific models. Given the cost and resources required to train such a model, the ultimate goal is to make available the techniques to replicate this work to stakeholders who work in aviation, so that they are able to train foundation models for future aviation and air traffic–related tasks. The techniques could also be applied outside of the aviation domain in other technical fields that use large amounts of domain-specific technical language.

XI. ACKNOWLEDGMENTS

Resources supporting this work were provided by the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing (NAS) Division at Ames Research Center.

REFERENCES

[1] He, Pengcheng, et al. "Deberta: Decoding-enhanced bert with disentangled attention." arXiv preprint arXiv:2006.03654 (2020).
[2] He, P., Gao, J. and Chen, W, "DeBERTaV3: Improving DeBERTa using electra-style pre-training with gradient-disentangled embedding sharing," arXiv preprint arXiv:2111.09543, 2021.
[3] Clark, K. "ELECTRA: Pre-training text encoders as discriminators rather than generators." arXiv preprint arXiv:2003.10555, 2020.
[4] Jacob. D., "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805, 2018.
[5] Nielsen, D., Clarke, S. and Kalyanam, K. "Towards an aviation large language model by fine-tuning and evaluating transformers." 43^{rd} AIAA/IEEE Digital Avionics Systems Conference (DASC), San Diego, 2024.
[6] Matthews, Bryan, Immanuel Barshi, and Jolene Feldman. "An approach to identifying aspects of positive pilot behavior within the aviation safety reporting system." 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC). IEEE, 2023.
[7] Chandra, Chetan, et al. "Aviation-BERT: A preliminary aviation-specific natural language model." AIAA AVIATION 2023 Forum. 2023.
[8] Wang, Liya, et al. "AviationGPT: A large language model for the aviation domain." AIAA AVIATION FORUM AND ASCEND 2024. 2024.
[9] Zhu, Yukun, et al. "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books." Proceedings of the IEEE international conference on computer vision. 2015.
[10] Wang, Alex, et al. "GLUE: A multi-task benchmark and analysis platform for natural language understanding." arXiv preprint arXiv:1804.07461 (2018).
[11] Beltagy, Iz, Kyle Lo, and Arman Cohan. "SciBERT: A pretrained language model for scientific text." arXiv preprint arXiv:1903.10676 (2019).
[12] Saxe, Andrew M., James L. McClelland, and Surya Ganguli. "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks." arXiv preprint arXiv:1312.6120 (2013).
[13] Wei, Yucheng, et al. "Analysis of Random Initialization Methods in Machine Learning." Proceedings of the 2024 Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence. 2024.
[14] Narkhede, Meenal V., Prashant P. Bartakke, and Mukul S. Sutaone. "A review on weight initialization strategies for neural networks." Artificial intelligence review 55.1 (2022): 291-322.
[15] Makkuva, Ashok Vardhan, et al. "Local to global: Learning dynamics and effect of initialization for transformers." Advances in Neural Information Processing Systems 37 (2024): 86243-86308.
[16] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[17] “Innovative Technology use in the Extraction of Flight Constraints recorded in Letters of Agreement (LOA).” International Civil Aviation Organization Assembly — 41st Session, 2022, https://www.icao.int/Meetings/a41/Documents/WP/wp_496_en.pdf