Naval Information Warfare Center PACIFIC

TECHNICAL REPORT 3257
JANUARY 2022

Applications of Natural Language Processing to Predict Components of Naval Aviation Readiness

Dr. Andrew B. Sabater
Dr. Benjamin A. Michlin
Josh Duclos
Gary R. Williams
Dean Lee
Dr. Jamal Rorie
NIWC Pacific

DISTRIBUTION STATEMENT A: Approved for public release. Distribution is unlimited.

Naval Information Warfare Center Pacific (NIWC Pacific)
San Diego, CA 92152-5001

This page is intentionally blank.

TECHNICAL REPORT 3257
JANUARY 2022

Applications of Natural Language Processing to Predict Components of Naval Aviation Readiness

Dr. Andrew B. Sabater
Dr. Benjamin A. Michlin
Josh Duclos
Gary R. Williams
Dean Lee
Dr. Jamal Rorie
NIWC Pacific

DISTRIBUTION STATEMENT A: Approved for public release. Distribution is unlimited.

Administrative Notes:

This report was authored in August 2021, approved through the Release of Scientific and Technical Information (RSTI) process in September 2021 and formally published in the Defense Technical Information Center (DTIC) in January 2022.

NIWC Pacific
San Diego, CA 92152-5001

NIWC Pacific
San Diego, California 92152-5001

A. D. Gainer, CAPT, USN
Commanding Officer

W. R. Bonwit
Executive Director

ADMINISTRATIVE INFORMATION

The work described in this report was performed by the Nonlinear Dynamics and Materials Research Branch (71780) of the Basic and Applied Research Division (71700), Naval Information Warfare Center Pacific (NIWC Pacific), San Diego, CA. The NIWC Pacific Naval Innovative Science and Engineering (NISE) Program provided funding for this Rapid Prototyping project.

Released by
John deGrassie, Division Head
Basic and Applied Research

Under authority of
Carly Jackson, Department Head
Cyber, Science and Technology

ACKNOWLEDGMENTS

This is a work of the United States Government and therefore is not copyrighted. This work may be copied and disseminated without restriction.

The citation of trade names and names of manufacturers is not to be construed as official government endorsement or approval of commercial products or services referenced in this report.

Editor: MRM

EXECUTIVE SUMMARY

DARTE (Digital Aviation Readiness Technology Engine) is a family of artificial intelligence and machine learning models that are used to predict Naval aviation readiness. With the goal of improving DARTE predictions, this document explored means to integrate text data into the modeling using natural language processing (NLP). To provide context, basic information on NLP as well as common methods for document classification were introduced. Based on the needs of DARTE, the Doc2Vec algorithm was selected as it provides a means to produce dense, constant-length numerical representations of documents. Background information on the foundations of Doc2Vec as well as the key hyperparameters are discussed. A variety of models of different complexity were implemented and tested. As the number of features added to the model increased, it was found that the relative importance of the Doc2Vec features decreased. Work was then conducted and connections were found between the Doc2Vec features and current features in the DARTE models. Many features that are known to be important components of DARTE's predictions can be accurately classified using Doc2Vec, but there is significant dispersion in how this information is encoded in the text data. Current efforts are focused on understanding this. Future efforts may also explore different NLP techniques to identify unique or unknown features useful for DARTE.

This page is intentionally blank.

CONTENTS

FIGURES

TABLES

1. INTRODUCTION AND MODELING GOALS

DARTE (Digital Aviation Readiness Technology Engine) is a family of artificial intelligence and machine learning models that are used to predict Naval aviation readiness [1, 2]. Readiness is forecasted by predicting the number of mission capable (MC) aircraft that a squadron will have and their quarterly flight hour execution. To make these predictions, DARTE utilizes numerical and categorical data aggregated from multiple sources. These data sources also contain free-form text. For example, for each maintenance event, a maintenance action form (MAF) is filled out and contains text related to a description of the issue and the corrective actions taken to address said issue. Given the direct relationship between this text data and maintenance events, it was hypothesized the inclusion of text data with DARTE could aid in predicting readiness. This work presents efforts related to testing this idea as well as background on means to utilize and process text data.

This page is intentionally blank.

2. INTRODUCTION TO NATURAL LANGUAGE PROCESSING

In a very general sense, natural language processing (NLP) is a means for automated or computerized systems to handle text data [3]. With such a broad definition, NLP can be used for very simple tasks like tokenization (segmentation of works from sentences) to more complicated tasks like question and answer systems and document paraphrasing. Expertise in NLP utilizes knowledge in both linguistics and computer science. NLP has come in three waves of innovation. The first wave was rule-based approaches developed by subject matter experts. These systems are still capable of excellent performance, but can require a significant amount of time to tune and optimize. The second wave was statistical or machine learning approaches to NLP. The main difference with the second wave was that instead of an expert determining the rules for a given NLP task, an algorithm would learn them based on a given corpus or collection of texts and documents. The third and current wave of NLP is the artificial intelligence or neural network approach to NLP. While similar to the second wave, one of the main advantages is that intermediate NLP tasks can be layered which provides an end-to-end means for optimization. One disadvantage of utilizing artificial intelligence for NLP is that it can require a large amount of data. [1] However, with the growth of available digital text in an easily accessible repository (i.e., the Internet), this has become less of an issue for certain applications.

[1]To provide context for what constitutes large data, in [6], a subset of 783 million tokens from the then Google News corpus of 6 billion tokens was used to train Doc2Vec models.

This page is intentionally blank.

3. INTRODUCTION TO DOC2VEC

With the goal of providing a means to utilize free-form text to improve the predictions of DARTE, methods for document classification were explored. One of the most common statistical approaches is termfrequency - inverse document frequency (TF-IDF), which for a given document, returns a vector [4].

TF-IDF belongs to the "bag of words" approaches where the order of words and semantics is ignored. To calculate the TF-IDF of a document, for each token or word in the document, one multiples the term frequency (TF) by the inverse document frequency (IDF). The TF is simply the number of times a token appears in a document divided by the number of tokens in the document. The idea being similar documentshave similar term frequencies. To counter the effect of certain words that appear repeatably and convey little information (e.g. "a", "the", "and", "of", etc.), the IDF is calculated to penalize these terms. A common formulation for IDF is $log(N)/df$, where N is the total number of documents in a corpus and df is the number of documents that contain a given token. For a given token and corpus, IDF is a constant. After the TF-IDF of a document is calculated, to account for documents of different lengths, the document's vector can be compared to others via cosine similarity. If the vectors have been L2-normalized,a simple dot product can be used.

There are several downsides with using TF-IDF for certain NLP tasks. The first is that the length of the document vector will be equal to the number of tokens in the corpus. The next is that the model has to be retrained for every new token that is added to the corpus. Lastly, information related to the ordering of words and semantics is ignored. While bi-grams or n-grams can be created to preserve some semantic information, this further exacerbates the sparsity issue of TF-IDF. To overcome some of the issues with TF-IDF, in 2014 Le and Mikolov developed Doc2Vec [5]. To understand Doc2Vec, background on their prior algorithm Word2Vec is helpful [6, 7].

The goal of the Word2Vec algorithm is to create word vectors in an unsupervised method (i.e. unlabeledfree text) that preserve that words can have multiple degrees of similarity. Below are some examples of testrelationships that, prior to Word2Vec, Mikolov et al. utilized [8].

- "Athens" is to "Greece" as "Madrid" is to "Spain"
- "man" is to "woman" as "nephew" is to "niece"
- "warm" is to "warmest" as "good" is to "best"

Put another way, the goal is to create a system that can learn word analogies. To assess the word vectorsand the learned relationships, Mikolov et al. proposed a simple mathematical formulation of word analogies “A” is to “B” as “C” is to “D” as

$$A - B + C \approx D, \tag{1}$$

where *A*, *B*, *C*, and *D* represent the respective word vectors. The approximately equal is used as it represents the attempt to find the closest word in the utilized corpus. The most famous example in their work is “King” − “Man” + “Woman” ≈ “Queen”. This arithmetic could also be used to assess singularand plural relationship (“apple” − “apples” ≈ “family” − “families”) or the tenses of words.

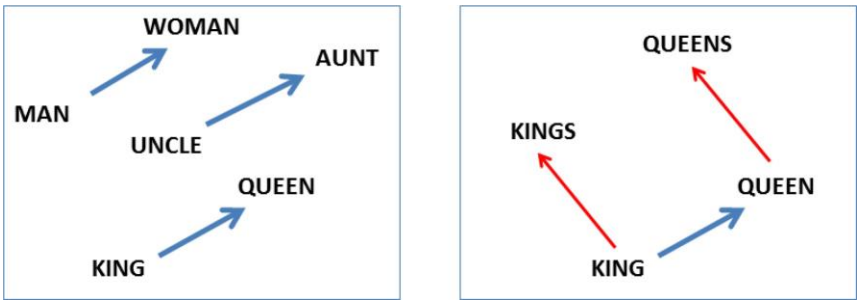

Figure 1. Sample words and a projection of their word vectors to a two-dimensional space [8].

Mikolov et al.’s work prior to Word2Vec that developed the framework to test word vectors utilized a recurrent neural network model [8]. His following work proposed to two log-linear models: continuous bagof words (CBOW) and continuous skip-gram [6]. For the NLP tasks described in that work, both Word2Vecmodels outperformed the recurrent neural network model. For CBOW, the order of words does not matter and the word vectors are effectively averaged. However, words before and after the word to be predicted are utilized. Thus, unlike standard bag of words models, a continuous distributed representation is utilizedfor the context (the standard being all words in the document are utilized for the context). The continuous skip-gram aims to predict words within a given range before and after the input. The continuous skip-gram model assumes that distant words are less related than closer words, so further words are weighed less bysampling them less.

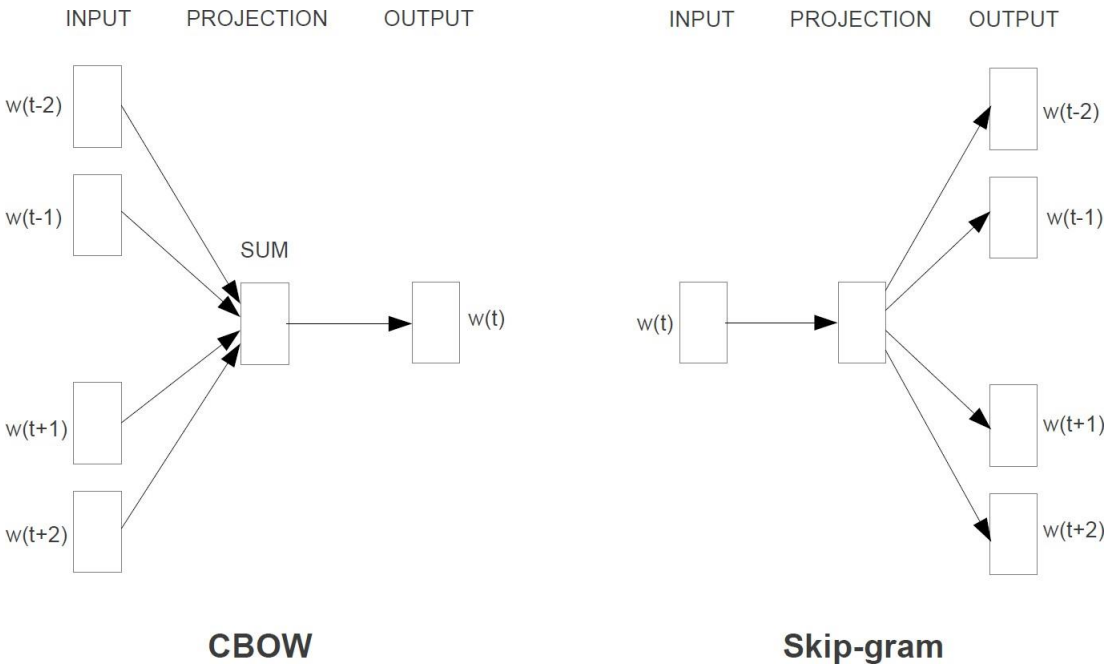

Figure 2. Model architectures for CBOW and skip-gram models [6].

At the time Word2Vec was developed, an end application was not envisioned. To quote from the first paper on Word2Vec "(w)ord vectors with such semantic relationships could be used to improve many existing NLP applications, such as machine translation, information retrieval and question answering systems, and may enable other future applications yet to be invented" [6]. To utilize the Word2Vec framework for text such as phrases, sentences, paragraph, etc., several groups proposed using averaged word vectors or using the word vectors as input to another model. However, Mikolov et al. claimed that the separate training of word vectors from other vectors used to encode longer texts produced poor results. Instead, Mikolov et al. proposed the Doc2Vec approach for paragraph vectors that builds on Word2Vec [5].

Building upon CBOW, the paragraph vector - distributed memory (PV-DM) model aims to predict words given input words and the context of a given paragraph. During training, word vectors are shared, but paragraph vectors are unique to each paragraph. Building upon the skip-gram model, paragraph vector - distributed bag of words (PV-DBOW) aims to predict words in a small window. Unlike PV-DM, the word vectors do not have to be learned and the context of surrounding words are ignored. Mikolov et al. note that PV-DM performs best most of the time, but they recommend a combination of PV-DM/PV-DBOW via concatenation of the vectors [5]. However, based on the numerical experiments presented in this work, PV-DBOW performs better.

The family of Doc2Vec algorithms have several hyper parameters that are used to tune model performance. For PV-DM, the context words can be either averaged together or concatenated. For PV-DBOW, either the word embeddings can be learned or they can be initialize to random values. Both models have a window size parameter. For computational reasons, instead of using a full softmax at the prediction layer, it can be simplified to either a hierarchical softmax or use negative sampling. For PV-DBOW, the objective function seeks to maximize the log probability of predicting words within the selected window given an input paragraph vector. While hierarchical softmax approximates the objective function of the full softmax, negative sampling does not. With negative sampling, words outside of the

selected window are randomly selected to decrease the objective function. To reduce the impact of very frequent words, often a sub-sampling parameter is specified that removes frequent words from the corpus. This can be important for a very large corpus, but was not shown to be useful for the numerical experiments conducted for this work. Lastly, as with most neural network models, the number of epochs, or iterations the objective function is perturbed to optimize, must be specified.

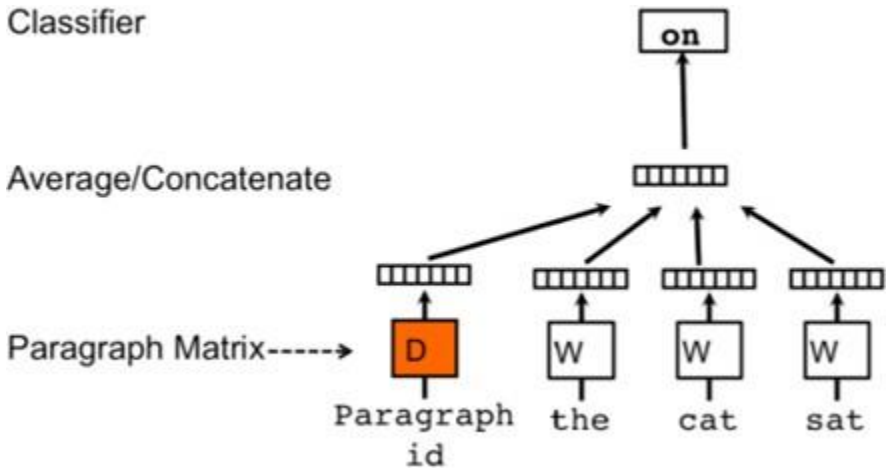

Figure 3. Model architecture for paragraph vector - distributed memory (PV-DM) [5].

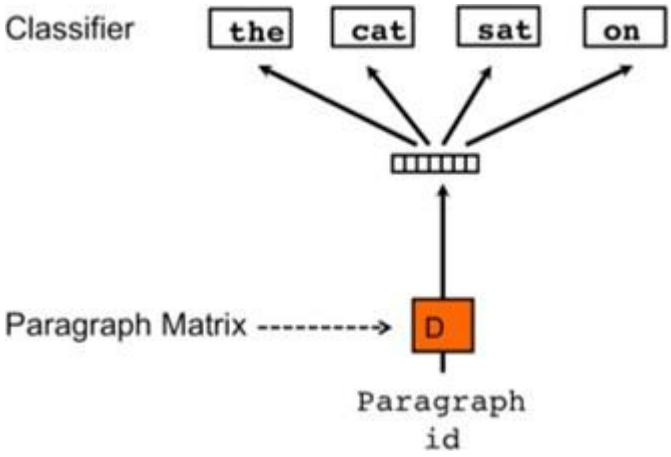

Figure 4. Model architecture for paragraph vector - distributed bag of words (PV-DBOW) [5].

4. DATA SOURCES

Several different data sources were utilized within this effort. The MC status of a given aircraft is typically reported in Aviation Maintenance Supply Readiness Report (ASMRR) and can change on a dailybasis. The current version of DARTE aims to predict MC several months in the future, so data from ASMRR was downsampled to monthly values. There are several categories used to describe the MC of a given aircraft. For example, an aircraft may not be mission capable due to a supply issue; this aircraft would be categorized as “NMCS”, a variation of not MC. For the purposes of building and testing simpleMC classifiers, MC was treated as a binary value. As the proportion of MC aircraft to not MC aircraft is not equal, issues associated with class balance were considered. The source of text data utilized are from Maintenance Action Form (MAF) records. Several fields associated with numerical or categorical data are available, but most work has been focused on the use of the text fields “corrective actions” and “description narrative.” To develop models that either predict or classify on monthly time scales, for a given aircraft and month, all of the corrective actions and descriptive narrative fields were aggregated into a single document. Other numerical and categorical data from MAFswere also used in this study. For example, the manhours spent on maintenance events for a given aircraft and month were aggregated to create a model feature. MAF records from fiscal years 2019 and 2020 wereused in this study. To give a sense of scale, more than 1.7 million maintenance events are documented in these forms. When the text data from these records were aggregated by a given aircraft and month to createdocuments, more than 10,000 were produced. As noted the following sections, these documents may be filtered to remove, for example, aircraft in a certain phase or squadron.

This page is intentionally blank.

5. BINARY CLASSIFIER AND HYPERPARAMETER ASSESSMENT

To study the impact of different model architectures and hyperparameters, a binary classifier for MC based on paragraph vectors trained on the aforementioned text data was trained. The Doc2Vec algorithms utilized were implemented in Gensim 4.0. Paragraph vectors of length 200 with 10 negative samples wereused. Three Doc2Vec models were trained, PV-DBOW, PV-DM where the context is averaged, and PV-DMwhere the context was concatenated. For brevity, the last two models are denoted as PV-DM/M and PV-DM/C, respectively. For PV-DBOW, a window size of 5 was used while both PV-DM models used window sizes of 10. Two additional models that are based on concatenated paragraph vectors were tests, denoted as PV-DBOW + PV-DM/M and PV-DBOW + PV-DM/C. To mitigate class imbalance issues, the balanced random forest classifier algorithm was used. Model performance was assessed based on F1 score.Both F1 micro and macro scores were calculated to assess model performance relative to class balance. The macro score is the average of the F1 score for both classes and the micro score accounts for class balance. five-fold cross validation was used where the reported value is the average of the five trials and theuncertainty was estimated as the standard deviation of the trials.

Table 1. MC Classifier Results for different Doc2Vec models.

Model	F1 Micro	F1 Macro
PV-DBOW	0.77 +/- 0.01	0.69 +/- 0.01
PV-DM/M	0.72 +/- 0.03	0.61 +/- 0.03
PV-DM/C	0.56 +-/ 0.08	0.53 +-/ 0.06
PV-DBOW + PV-DM/M	0.60 +/- 0.17	0.56 +/- 0.14
PV-DBOW + PV-DM/C	0.68 +/- 0.09	0.63 +/- 0.06

Of the tested models, PV-DBOW performed the best. Based on a simple numerical experiment, PV-DBOW based just on MAF text data predicts MC better than either a random guess or setting all predictions to the majority class. Thus, this would imply the model is able to learn some information about MC status based on text data. Interestingly, the concatenated models did not perform the best.

Following that experiment, another one was conducted with adding additional features to the model. Specifically, the following features, encoded as numerical values, were added to the model: CNAL designation, deployment phase, operational status, binned monthly MAF manhours, and prior MC status. With just a few added features, all models were significantly improved. Again, PV-DBOW performed the best.

According to [9], as an alternative to using random word embeddings, it is claimed that PV-DBOW performs better when either the word embeddings are trained at the same time or pretrained word embeddings are used.

Table 2. MC Classifier Results for different Doc2Vec models with additional features.

Model	F1 Micro	F1 Macro
PV-DBOW	0.79 +/- 0.01	0.73 +/- 0.01
PV-DM/M	0.73 +/- 0.09	0.67 +/- 0.06
PV-DM/C	0.69 +-/ 0.08	0.64 +-/ 0.07
PV-DBOW + PV-DM/M	0.65 +/- 0.17	0.61 +/- 0.15
PV-DBOW + PV-DM/C	0.76 +/- 0.03	0.70 +/- 0.01

For the case that word embeddings are not learned, which is the default, this is still designated as PV-DBOW. For the case that word embeddings are learned, the model is designated as PV-DBOW+W. Results are shown in Table 3. Considering the uncertainty of the results, there does not seem to be a significant difference between the models. However, PV-DBOW+W took approximately seven times longer to train and test. Thus, while it is possible that for certain application, learning the word embeddings or using pretrained word embeddings may be important or useful for some applications, it does not appear to be useful for this one.

Table 3. MC Classifier Results for PV-DBOW without and with learning word embeddings.

Model	F1 Micro	F1 Macro
PV-DBOW	0.789 +/- 0.006	0.728 +/- 0.004
PV-DBOW+W	0.788 +/- 0.010,	0.727 +/- 0.014

To quote [9], “(p)ossibly the most important hyper-parameter is the sub-sampling threshold for high frequency words: in our experiments we find that task performance dips considerably when a sub-optimal value is used.” They argue that subsampling accelerates learning and potentially improves the word vectorsof rare words. Connections have also been made between subsampling and the IDF weighting for TF-IDF approaches. In case that learning the word embeddings becomes more important with subsampling, for these trials word embedding were calculated. Results are shown in Table 4. As all of the results from thisexperiment are similar, as with the last hyper-parameter, it does not seem to sensitively impact the model.

Table 4. MC Classifier Results for PV-DBOW+W for different values of subsampling (t).

Model	F1 Micro	F1 Macro
PV-DBOW+W (t = 1e-5)	0.799 +/- 0.007	0.727 +/- 0.010
PV-DBOW+W (t = 1e-4)	0.793 +/- 0.007	0.730 +/- 0.010
PV-DBOW+W (t = 1e-3)	0.786 +-/ 0.009	0.726 +-/ 0.011

Numerical experiments were conducted to determine the impact of window size. According to [9], typically PV-DBOW generally favors longer windows than PV-DM. The default window size for gensim Doc2Vec implementation is 5, so experiments were conducted with window sizes of 5, 10, 20, 40, 80, 160, and 320. Results are shown in Table 5. As with the last experiment, due to the uncertainty in F1 scores, it is difficult to determine an optimal window length. Given the computational benefits of shorter windows and their similar performance to longer windows, in general short windows were used in the following experiments.

The last hyper-parameter that was explored was vector length. Results are shown in Figure 5. As the length of the vector increased, the F1 score appears to converge to particular value. One potential caveat is that longer vectors may need more epochs to train. Other numerical experiments where the number of epochs was increased from 10 to 100 did not significantly change model performance. Based on the results from this experiment, the vector length in following experiments was set to at least 25.

Following experiments related to tuning hyper parameters, efforts focused on improving the classifier. As prior models for DARTE utilized significantly more features, additional numerical and categorical features were added to the current model.

Table 5. MC Classifier Results for PV-DBOW for different window lengths.

Model	F1 Micro	F1 Macro
PV-DBOW (window=5)	0.791473 +/- 0.005138	0.730085 +/- 0.007419
PV-DBOW (window=10)	0.790763 +/- 0.005538	0.729700 +/- 0.007146
PV-DBOW (window=20)	0.788434 +-/ 0.006269	0.726991 +-/ 0.008383
PV-DBOW (window=40)	0.790865 +-/ 0.005112	0.729852 +-/ 0.007270
PV-DBOW (window=80)	0.788942 +/- 0.008961	0.727225 +/- 0.010154
PV-DBOW (window=160)	0.790867 +/- 0.011893	0.729240 +/- 0.013073
PV-DBOW (window=320)	0.790057 +-/ 0.012279	0.728478 +-/ 0.013816

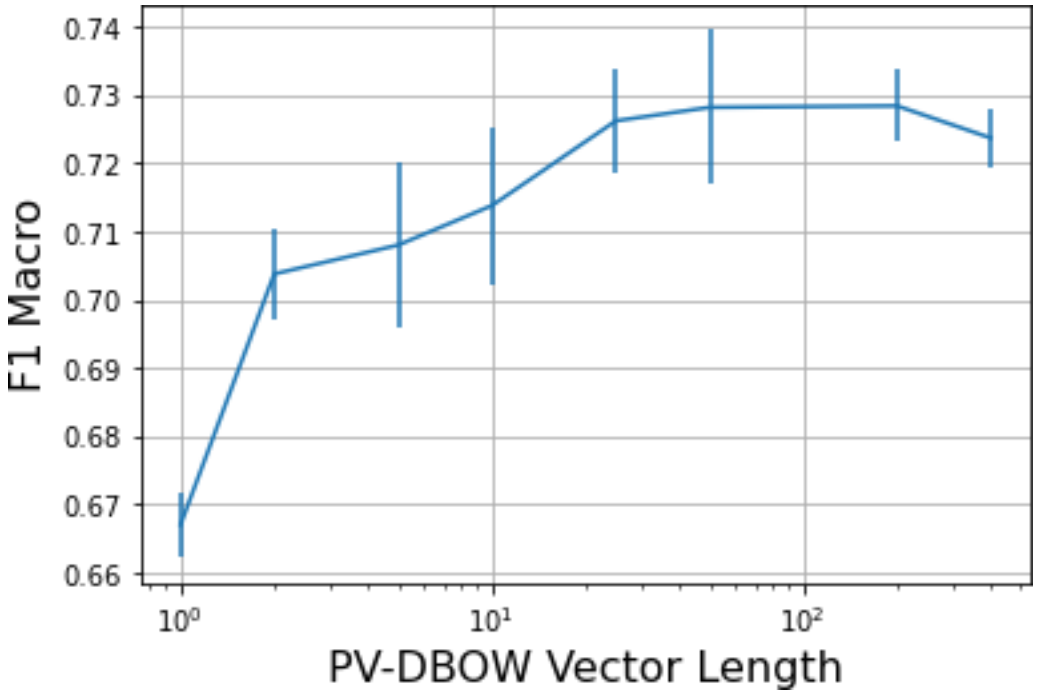

Figure 5. MC Classifier Results for PV-DBOW for different vector lengths.

To assess the impact of additional features to an MC classifier that utilizes Doc2Vec to process MAF text data, five models were proposed. Results are shown in Table 6.The models are described below:

- Baseline: Uses categorical and numerical features with more than one value and less than 10 unique values from one, two, and three months prior. Label encoding (not one-hot) was used. Total of 5,452 features.
- Baseline with NLP: Same features as prior model with MAF text data used to generate paragraph vectors.
- Baseline+ with NLP: 195 categorical features label encoded, 10,504 numerical features, and MAF text data.
- One-Hot Baseline+ with NLP (25): Categorical features from Baseline+ model one-hot encoded,same numerical features with same encoding as Baseline+ model, and paragraph vectors of length25.
- One-Hot Baseline+ with NLP (400): Same as prior model but with paragraph vectors of length 400.
- Clean One-Hot Baseline+ with NLP (25): Same numerical/categorical features prior model, but with text cleaning applied to text data. Paragraph vector length limited to 25.

Compared to prior models, the baseline model performs slightly better than the simple models with paragraph vectors. The addition of more features to the Baseline+ model provides the greatest relative improvement, particularly for the not MC/FMC class. Feature encoding, one-hot versus label, does not seem to make a significant difference.

While in general, text cleaning or pre-processing tends to improvemost NLP tasks, it is difficult to state the impact of text cleaning. For the presented numerical experiment,the following steps were taken for text cleaning:

- Make all text lower case
- Remove punctuation and excess white space
- Remove numerical values or text containing numerical values
- Remove stop words

  Default stops from Gensim used

  Based on word frequency, the following words were also removed:
  'fod','free','secure','area','ac','replaced','ataf','removed','pem a', 'received','aircraft','used','iaw','ietm','ietms','cell'
- Rules-based Porter stemming as implemented in Gensim

While the above is fairly standard text cleaning, the removal of text containing numerical values and certain words deemed to be stop words may be problematic. Eluding to future work, important informationsuch as squadron, would be removed. Text cleaning helped to reduce the size of the corpus and thus reducemodel training time, but at this time there is no clear benefit.

For these experiments, the addition of a paragraph vector, regardless of length, does not seem to significantly improve the model. To help quantify this, SHAP feature importance was utilized. Shown in Figure 6 is the sorted feature importance of the paragraph vector components for Clean One-Hot Baseline+ with NLP. Several components are of much greater importance than the rest, but after a step drop, the importance tends to decrease monotonically. However, when compared to the rest of the model features, thecontribution of paragraph vectors is rather small. Shown in Figure 7 is the feature importance of all features inthe model. The approximate location of the top Doc2Vec feature is shown with the red arrow. While the Doc2Vec features are within the top hundreds, the overall contribution to the model is relatively small.

While the relative feature important of the components of the paragraph vectors are small, it is interesting to note that the similar performance of the Baseline model to the simple models with paragraphvectors potentially infers that similar information is encoded in the MAF paragraph vectors. This in explored more in depth in the following section.

This page is intentionally blank.

6. INFORMATION ENCODED IN MAF PARAGRAPH VECTORS

Based on the results from the last sections, the addition of MAF-based paragraph vectors to a simple model are able to improve the prediction of MC status. However, as more features were added the model, the importance of the paragraph vectors were found to be relatively small.

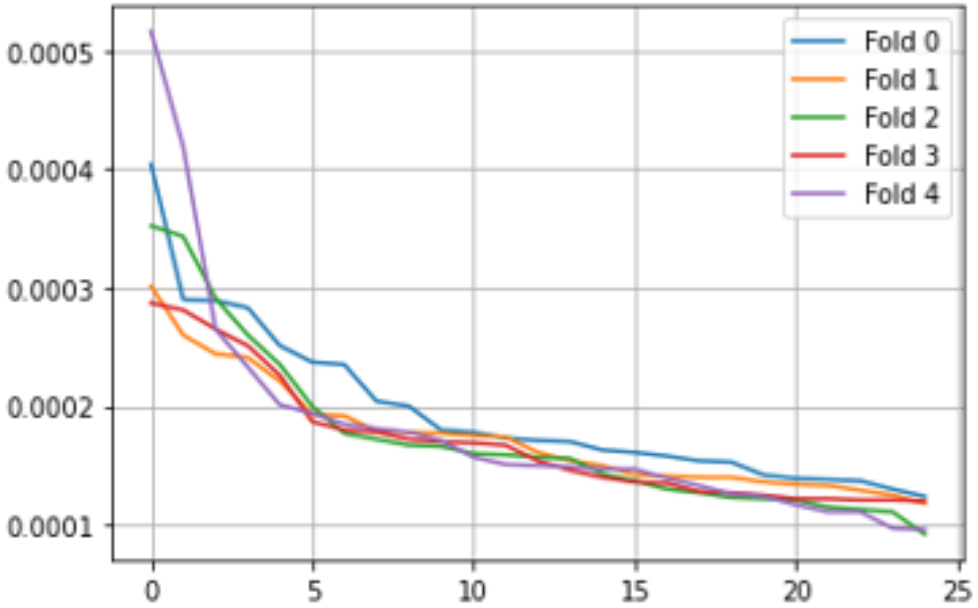

Figure 6. Sorted SHAP feature importance of paragraph vector components for Clean One-Hot Baseline+ with NLP.

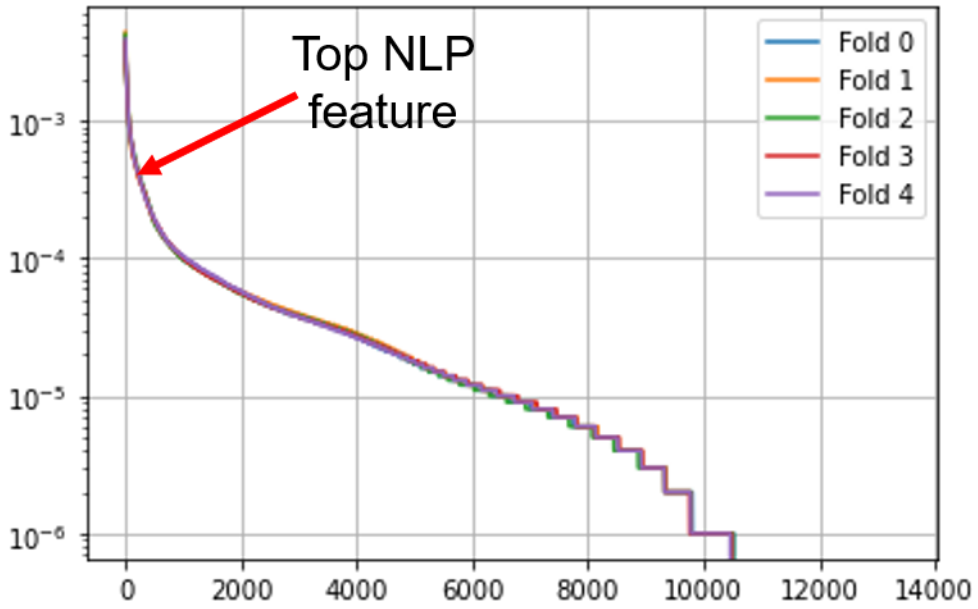

Figure 7. Sorted SHAP feature importance for Clean One-Hot Baseline+ with NLP.

Thus it was hypothesized thatthe information from the additional features may be encoded in the paragraph vectors. To assess this, the following test was implemented:

- Aggregate monthly MAF text data as discussed in earlier sections.
- Exclude or drop samples if the phase value is not populated or the squadron is VFA-106 or VFA-122. [2] The total number of samples of 8,932.

Table 6. MC Classifier Results for PV-DBOW for different feature spaces.

Model	F1 (MC/FMC)	F1 (Not MC/FMC)
Baseline	0.869882 +/- 0.007680	0.592126 +/- 0.016054
Baseline with NLP	0.883009 +/- 0.005956	0.598725 +/- 0.017522
Baseline+ with NLP	0.876282 +/- 0.002948	0.611403 +/- 0.013507
One-Hot Baseline+ with NLP (25)	0.871481 +/- 0.005362	0.610572 +/- 0.012812
One-Hot Baseline+ with NLP (400)	0.874101 +/- 0.009256	0.609166 +/- 0.026145
Clean One-Hot Baseline+ with NLP	0.870336 +/- 0.008315	0.605277 +/- 0.017751

- Train PV-DBOW model with vector length 100 on all text data.
- Train a balanced random forest classifier to predict categorical features for a given month and aircraft. Using 5-fold cross validation for train-test splits, assess model performance based on F1 score for each class.

Results of the experiment are shown in Figures 8 to 14. The length of the bars in the plots are the average values of the F1 scores, or the F1 macro score. The red dots are the F1 values for each class. The large yellow dot denotes the difference between the maximum and minimum F1 scores. The data in the plots wassorted based on F1 macro score. For many of the these categorical features, the F1 scores are very closely equal to one. Yet, based the current analysis of the MAF text data, words, tokens, or n-grams that explicitlyencode these features are not present at a very high frequency. This implies that the descriptive narrative and corrective actions of MAFs are recorded generally in specific manners based on location, operational status, squadron, variant, etc. This is very consistent for certain categorical features. For example, for location (denoted as “amsrr.acc”), the F1 scores for LANT NAVY and PAC NAVY are 0.98 and 0.99. For some, there is a rather wide range. For squadron, the average F1 score is 0.93, but the difference between the maximum and minimum F1 score is 0.18. Thus, while a significant amount of information in encoded in the MAF-based paragraph vectors, due to the dispersion of how well this information is encoded, they might not be a suitable replacement for this information in a model. Current efforts are focused on determining unique information related to MC status in the paragraph vectors and understanding the conditions or reasons for the dispersion in how well information is encoded in paragraph vectors.

[2]The noted squadrons are associated with training, so their phase information does not have the same meaning as other ones.

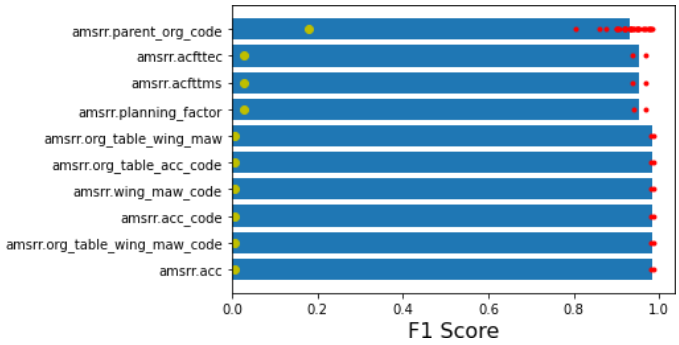

Figure 8. First F1 plot for classification of categorical features using MAF-based paragraph vectors.

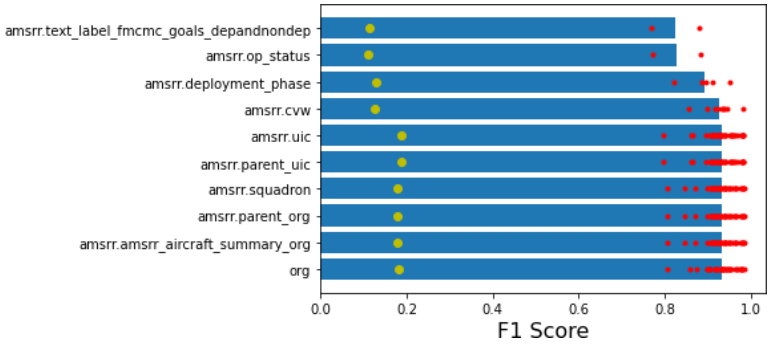

Figure 9. Second F1 plot for classification of categorical features using MAF-based paragraph vectors.

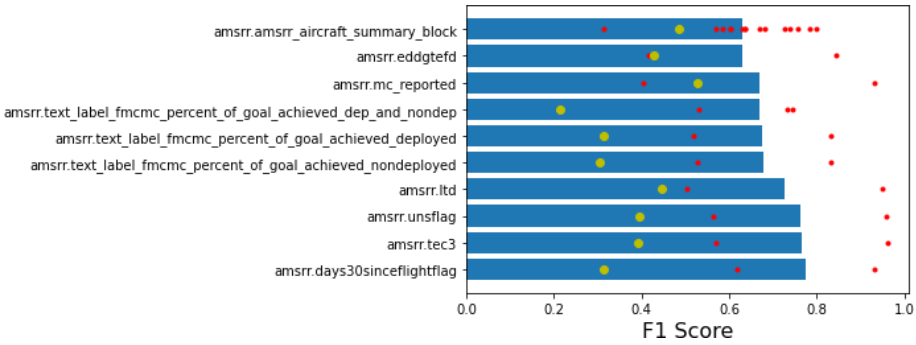

Figure 10. Third F1 plot for classification of categorical features using MAF-based paragraph vectors.

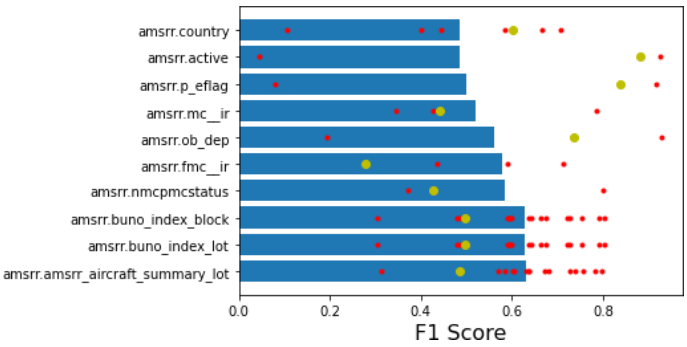

Figure 11. Fourth F1 plot for classification of categorical features using MAF-based paragraph vectors.

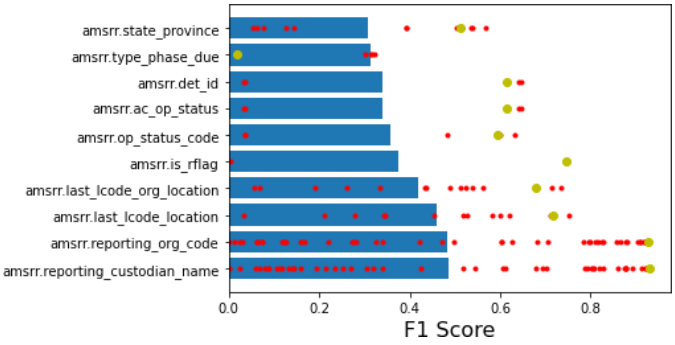

Figure 12. Fifth F1 plot for classification of categorical features using MAF-based paragraph vectors.

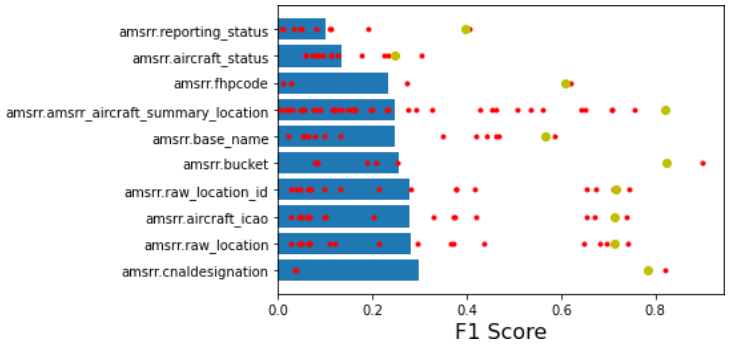

Figure 13. Sixth F1 plot for classification of categorical features using MAF-based paragraph vectors.

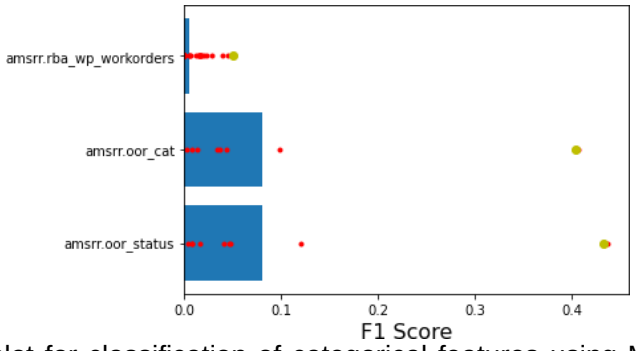

Figure 14. Seventh F1 plot for classification of categorical features using MAF-based paragraph vectors.

This page is intentionally blank.

7. CONCLUSION

With the goal of improving DARTE (Digital Aviation Readiness Technology Engine), work was conducted to determine if text data in maintenance action forms (MAFs) related to maintenance events could be used to improve readiness predictions. Background information related to natural language processing (NLP) as well as common NLP methods for document classification was given. Motivated by the need to have an unsupervised means to produce dense, constant-length numerical representations of documents, Doc2Vec was selected for this task. Information on key hyper-parameters for the generation ofthese paragraph vectors was assessed on actual data. A series of models of different complexity that utilizedDoc2Vec were trained and tested. Based on simple numerical experiments, within the family of Doc2Vec algorithms, the PV-DBOW (paragraph vector - distributed bag of words) model based just on MAF text data predicted mission capable (MC) status of aircraft better than either a random guess or setting all predictions to the majority class. Thus this would imply the model is able to learn some information aboutMC status based on text data. However, it was found that as the complexity of the model increased, the relative importance of Doc2Vec features decreased. Motivated by this, work was conducted to determine ifdata encoded in the paragraph vectors was present in the current datasets that prior DARTE models used. It was found that many key features for DARTE are encoded in MAF-based paragraph vectors. The of range of how well this information is encoded is broad, so current efforts are focused on studying this. Future efforts may explore the use of other NLP techniques to find new or helpful features for DARTE.

This page is intentionally blank.

REFERENCES

1. Michlin, B., Chang, R., Cruz, R., Duclos, J., Lee, D., Siu, V., and Yetman, C. 2019. "Predicting FA-18 squadron readiness and quarterly flight hour execution using machine learning," Tech. Rep. XXX, Naval Information Warfare Center Pacific.

2. Michlin, B., Duclos, J., Williams, G., Lee, D., Sabater, A. B., and Rorie, J. 2021. "Improving the Digital Aviation Readiness Technology Engine (DARTE) with Temporal Pattern Attention Mechanisms and Hyper-Deep Ensembles," Tech. Rep. XXX, Naval Information Warfare Center Pacific.

3. Eisenstein, J. 2018. *Natural Language Processing*, MIT Press, Cambridge, MA, 1st ed., URL https://github.com/jacobeisenstein/gt-nlp-class/blob/master/notes/eisenstein-nlp-notes.pdf.

4. Manning, C. D., Raghavan, P., and Schütze, H. 2009. *An Introduction to Information Retrieval*, Cambridge University Press, Cambridge, England, 1st ed., URL https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf.

5. Le, Q. and Mikolov, T. 2014. "Distributed Representations of Sentences and Documents," *31 st International Conference on Machine Learning*, URL https://arxiv.org/pdf/1405.4053.pdf.

6. Mikolov, T., Chen, K., Corrado, G., and Dean, J. 2013. "Efficient Estimation of Word Representations in Vector Space," *International Conference on Learning Representations 2013*, URL https://arxiv.org/pdf/1301.3781.pdf.

7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. 2013. "Distributed Representations of Words and Phrases and their Compositionality," *26th International Conference on Neural Information Processing Systems*, URL https://arxiv.org/pdf/1310.4546.pdf.

8. Mikolov, T., Yih, W., and Zweig, G. 2013. "Linguistic Regularities in Continuous Space Word Representations," *2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, URL https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/rvecs.pdf.

9. Lau, J. H. and Baldwin, T. 2016. "An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation," *1st Workshop on Representation Learning for NLP*, URL https://arxiv.org/pdf/1607.05368.pdf.

This page is intentionally blank.

INITIAL DISTRIBUTION

84310	Technical Library/Archives	(1)
71780	A. Sabater	(1)
53424	B. Michlin	(1)
71740	J. Duclos	(1)
53424	G. Williams	(1)
53424	D. Lee	(1)
53424	J. Rorie	(1)

Defense Technical Information Center
Fort Belvoir, VA 22060–6218 (1)

Naval Innovative Science and Engineering (1)

This page is intentionally blank.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-01-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden to Department of Defense, Washington Headquarters Services Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
January 2022	Final	

4. TITLE AND SUBTITLE

Applications of Natural Language Processing to Predict Components of Naval Aviation Readiness

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHORS

Dr. Andrew B. Sabater
Dr. Benjamin A. Michlin
Josh Duclos
NIWC Pacific

Gary R. Williams
Dean Lee
Dr. Jamal Rorie
NIWC Pacific

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NIWC Pacific
53560 Hull Street
San Diego, CA 92152–5001

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-3257

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

NIWC Pacific, NISE
53560 Hull Street
San Diego, CA 92152–5001

10. SPONSOR/MONITOR'S ACRONYM(S)

NISE

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release. Distribution is unlimited.

13. SUPPLEMENTARY NOTES

This is a work of the United States Government and therefore is not copyrighted. This work may be copied and disseminated without restriction.

14. ABSTRACT

DARTE (Digital Aviation Readiness Technology Engine) is a family of artificial intelligence and machine learning models that are used to predict Naval aviation readiness. With the goal of improving DARTE predictions, this document explored means to integrate text data into the modeling using natural language processing (NLP). To provide context, basic information on NLP as well as common methods for document classification were introduced. Based on the needs of DARTE, the Doc2Vec algorithm was selected as it provides a means to produce dense, constant-length numerical representations of documents. Background information on the foundations of Doc2Vec as well as the key hyperparameters are discussed. A variety of models of different complexity were implemented and tested. As the number of features added to the model increased, it was found that the relative importance of the Doc2Vec features decreased. Work was then conducted and connections were found between the Doc2Vec features and current features in the DARTE models. Many features that are known to be important components of DARTE's predictions can be accurately classified using Doc2Vec, but there is significant dispersion in how this information is encoded in the text data. Current efforts are focused on understanding this. Future efforts may also explore different NLP techniques to identify unique or unknown features useful for DARTE.

15. SUBJECT TERMS

Readiness; Naval Aviation Readiness; DARTE; Natural Language Processing; NLP; Word2Vec; Doc2Vec

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Andrew Sabater
U	U	U	SAR	40	19B. TELEPHONE NUMBER *(Include area code)* (619) 553-2480

This page is intentionally blank.

This page is intentionally blank.

DISTRIBUTION STATEMENT A: Approved for public release. Distribution is unlimited.

Naval Information Warfare Center Pacific (NIWC Pacific)
San Diego, CA 92152-5001