SYM-AM-20-052

PROCEEDINGS
OF THE
SEVENTEENTH ANNUAL
ACQUISITION RESEARCH SYMPOSIUM

Acquisition Research:
Creating Synergy for Informed Change

May 13–14, 2020

Published: April 10, 2020

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Defense Management at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Application of Natural Language Processing to Defense Acquisition Executive Summary Reports

Madison Hassler—is a Data Scientist at Rotunda Solutions. In her current role at OUSD(A-S), she provides data science consulting services to inform key leaders with data-driven solutions. Prior to her work with Rotunda, she worked as a civilian Data Scientist with the Air Force and used machine learning to improve their inventory models. She is passionate about creating understandable and interpretable products based on data for decision-makers to utilize. She has a master's degree in Systems Engineering from the University of Virginia and a bachelor's degree in Mathematics from Furman University. [madison.l.hassler2.ctr@mail.mil]

Terrence Clark—is a Senior Data Scientist at Rotunda Solutions. His experience bridges technology, data, and research, allowing him to create and manage solutions from inception to production. In his current role, he provides analytics consulting services to key leaders in the DoD OUSD(A-S). Additionally, he advises on data management, analysis, and infrastructure requirements for the DoD CIO, Tanzania dLab, the Joint Artificial Intelligence Center, and various private organizations. Some of his previous roles include price optimization, leading a corporate data science team at a Fortune 100, and developing predictive models for retail and restaurant clients. [terrence.j.clark5.ctr@mail.mil]

Abstract

Major Defense Acquisition Programs (MDAPs) are required to submit quarterly Defense Acquisition Executive Summary (DAES) reports which, among other information, contain ratings for each program area (green, yellow, red, etc.) and explanations of these ratings by the program manager. Natural language processing, a powerful machine learning tool, can harness the wealth of text data available in these reports in order to predict the ratings given the program manager's explanation in the report. With this information, the model can be used to indicate which programs are not reporting their ratings as expected in order to indicate which programs may need further investigation. Utilizing machine learning in this manner can increase insights into data in the DAES reports and has broad implications for further applications of these techniques to other acquisition data.

Keywords

1. Military acquisition and procurement
2. Machine learning and artificial intelligence
3. United States Department of Defense

Introduction

The 2018 National Defense Strategy, as well as recent National Defense Authorization Acts (NDAAs), have called for increased use of artificial intelligence and machine learning in the Department of Defense. With the wealth of acquisition data available, there is ample opportunity to take advantage of the latest machine learning techniques to this end. Natural language processing of acquisition text data can improve the workflow within the Office of the Secretary of Defense (OSD) and increase insights into the data available. With thousands of DAES reports for the MDAPs, natural language processing can significantly reduce the number of man-hours required to review future reports. Natural language processing models can assist human analysts in identifying programs that need to be reviewed further.

Methodology

Natural language processing is a branch of artificial intelligence that uses natural human language and processes it in a way that computers can understand. Examples of

natural language processing include voice assistants such as Apple's Siri or Amazon's Alexa. These are computers that listen to the natural language of a human, process it for the data it contains, and respond with audio based on the information it gathers. Frequently, machine learning is applied to the data to gain insight from the input. Other typical applications include language translation with Google Translate or word processing with Microsoft Word or Grammarly to check the grammatical accuracy of the text.

Computers have a difficult time understanding the subtleties of language, which makes deriving meaning from language a problematic task. Many rules govern natural language. Some are trivial and others more abstract. For example, using "s" to designate a plural is a rather low-level rule, but sarcasm is much more subtle and difficult to detect. Natural language processing must apply algorithms to learn these rules so that computers can extract meaning from natural language data. Early efforts in natural language processing used a series of handwritten rules and algorithms to do this as more sophisticated computer algorithms were either unavailable or the technology to perform the computations was not available yet. Beginning in the 1980s, as computers improved, natural language processing was able to use more traditional machine learning models such as decision trees. The early aughts brought about the use of neural networks in natural language processing (Canuma, 2019).The idea of neural networks, otherwise known as deep learning, originated from a model of how neurons in the brain function and became famous as a method for image recognition. The techniques and theories of building neural networks methods had existed for quite a while, but with the advent of more powerful computers, the field rapidly expanded.

Sentiment analysis was one field of natural language processing that was able to expand with neural networks. Sentiment analysis is the process of detecting whether the sentiment of a given text is positive or negative. A classic example of this is using a database of movie reviews and determining whether the review is positive or negative. Machine learning algorithms can be used to extract features from a text and then predict the tag (positive, negative, etc.) that is associated with the text. A high-level overview of sentiment analysis is shown in Figure 1.

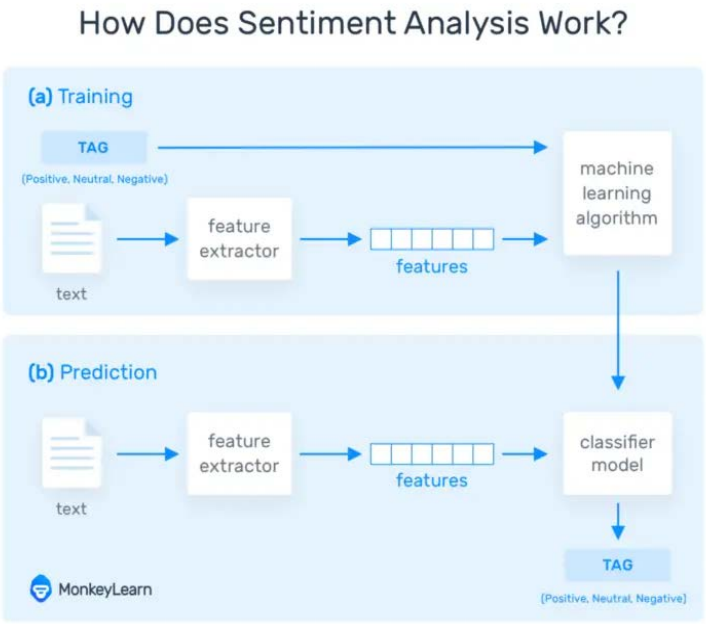

Figure 1. Image depicting how sentiment analysis is done. (Garbade, 2018).

Neural networks have allowed the expansion and success of word embeddings to take the forefront in natural language processing tasks such as sentiment analysis and have now been able to outperform more traditional machine learning models. Word embeddings are numeric representations of words that are used to predict a word based on its context, among other words. Words that have similar meanings should have a similar representation in the embedding. In 2013, Mikolov et al. introduced the word2vec embedding, which outperformed previous efforts to create a numerical representation of words (2013). This embedding uses neural networks to assess whether words appear in similar contexts and encodes this into the vector representation of a word. The most famous example of the word2vec embedding enabled a computer to learn the relation that "king" - "man" + "woman" = "queen." This embedding and other learned relationships are depicted in Figure 2.

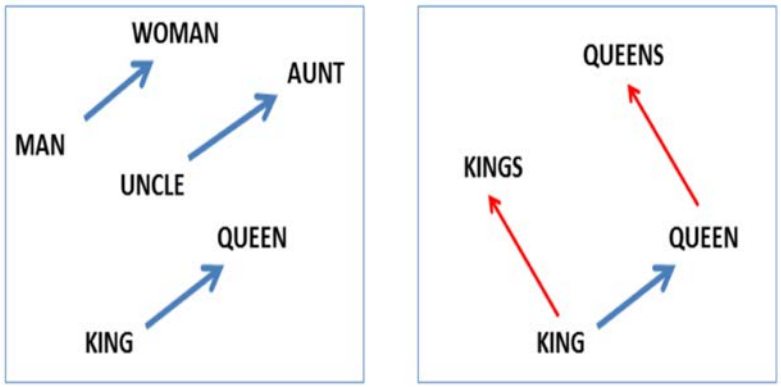

Figure 2. Image depicting some learned relationships of the word2vec embedding. (Mikolov, Yih, et al., 2013).

To create a word embedding, a generic representation of a word, typically initially using one-hot encoded vectors, is created. An example of one-hot encoding for the sentence “This is fun.” is shown below.

This = [1, 0, 0]

is = [0, 1, 0]

fun = [0, 0, 1]

These one-hot encoded vectors are then fed into a neural network and using various techniques are manipulated into less generic and more specific numeric representations based on the context of the word in different settings. A generic illustration of this for one word is shown in Figure 3.

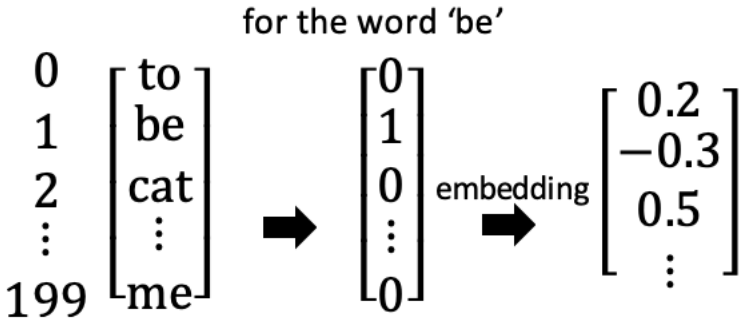

Figure 3. An overview of building an embedding for the word “be” from a one-hot-encoded vector

This embedding layer, or matrix of word representations, is then fed into the input of the neural network for prediction. There are several different types of neural networks. It has been shown that convolutional neural networks can have success with classification problems such as sentiment analysis (Dauphin, Fan, Auli, & Grangier, 2017; Gehring, Auli, Grangier, Yarats, & Dauphin, 2017). The architecture of a generic convolutional network used for classification is shown in Figure 4.

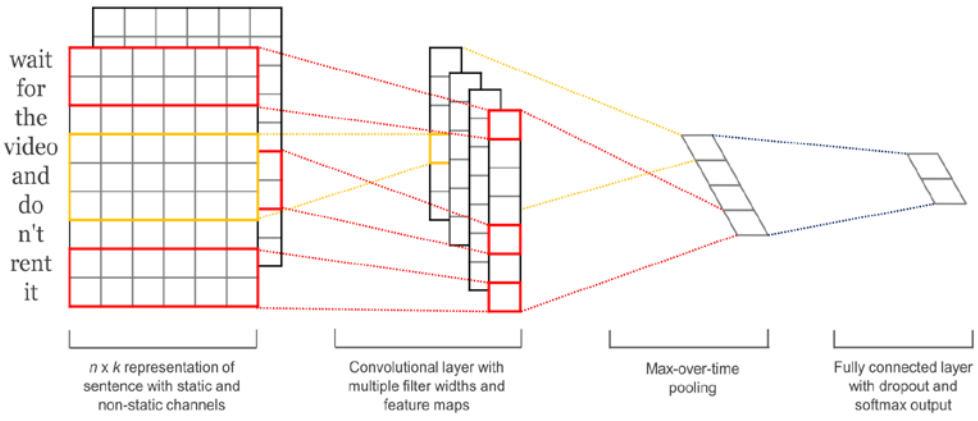

Figure 4. Image depicting a generic architecture for a convolutional neural network. (Zaman & Mishu, 2017).

For a classification problem such as sentiment analysis, the output layer predicts each class (e.g., positive, negative, etc.).

There are many applications of natural language processing, and here we have explored how deep learning can be used in sentiment analysis to predict sentiment classes. Further examples of where text classification is being used currently include market research, language detection, profanity, and abuse detection. Deep learning can be used to analyze product reviews for a company and compare it to reviews of a competitor in market research, or it can be used to categorize customer feedback into different topics. In social media, text classification is being used to determine the language of a given post. It is also being used to detect profanity and abuse to flag posts as bullying or hate speech and mark posts for removal. In addition to these existing applications, the field of deep learning for natural language processing is continually expanding. Other methods are using character-based as opposed to word-based models, which alleviates some issues with obscure words (Conneau, Schwenk, Barrault, & Lecun, 2016).

Acquisition data within OSD is a domain where previously natural language processing has not been fully explored. For example, the quarterly DAES reports include an assessment by the program manager of green, yellow, or red and includes an associated explanation for the rating. This is an example of where sentiment analysis could be applied. A model can be built that would be able to predict a program manager's rating and potentially be able to predict when a program's rating is likely to change. Additionally, natural language processing could be applied to these ratings to flag explanations for further evaluation for OSD analysts. Various other natural language processing tasks could also be explored with this text data set such as summarizing the explanations or classifying the explanation as a particular topic. These applications will be explored further in the next section. The advancement of natural language processing in recent years has made it a field that can be easily implemented in an organization, and it can provide value to decision making.

Case Study: DAES Program Manager Assessments

Each quarter, MDAPs are required to submit program assessments known as DAES. These reports represent pre-decisional assessments by the program manager in various program areas (e.g. cost, schedule, performance, etc.) to document areas of concern and advise leadership of potential emerging risks. Program managers assign one of three ratings to each program area (red, yellow, or green). It should be noted that in some cases a rating is not required to be reported by the program manager and in these cases the value "no rating" is supplied. Historically, advisory ratings have also been assigned (red advisory, yellow advisory, and green advisory). The data used for this model is from DAES reports ranging from 2001–2019. The primary fields of interest for this analysis are the program manager's rating and their explanation for the given rating. The text from the explanation field is used to build our natural language processing model.

To be able to analyze the explanation data, several text cleaning steps are taken. One step is to remove any trace of program or subprogram name from the explanation text. Programs do not tend to make large changes in ratings over time so rather than creating a model that would be very predictive for a program given the program name, we want to create a model that is generic for any program including programs not yet seen. Program and subprogram names were also split into separate word parts so that for the "F-35 Aircraft" subprogram both "F-35 Aircraft" and "F-35" are removed, for example, and replaced with either "programname" or "subprogramname" as appropriate. These were replaced intentionally as a single word instead of two so that it would be treated as only one piece of

information in the text. Rating names were also removed from the explanation text. Several explanations included something similar to “The program is rated green because …”. As this would be highly indicative of what the prediction should be, all instances of “green,” “yellow,” “red,” etc. are removed from the explanation text. Additionally, many of the texts contained HTML tags and character codes such as or &. These are typically used in the formatting and displaying of the text on the DAMIR website, but they are not useful for analysis; thus, the HTML tags were removed, and the character codes were replaced with the appropriate character. These steps were performed for the entire data set.

The DAES assessment data set contained approximately 213,000 data points. After removing entries where either the rating or the explanation is empty, approximately 43,000 data points were remaining. Data with explanations shorter than 35 characters were also removed as these contained no useful predictive data (i.e., “Program on track. Not applicable”). The remaining data set contained approximately 41,000 points. The class breakdown for this data set is shown in Figure 5.

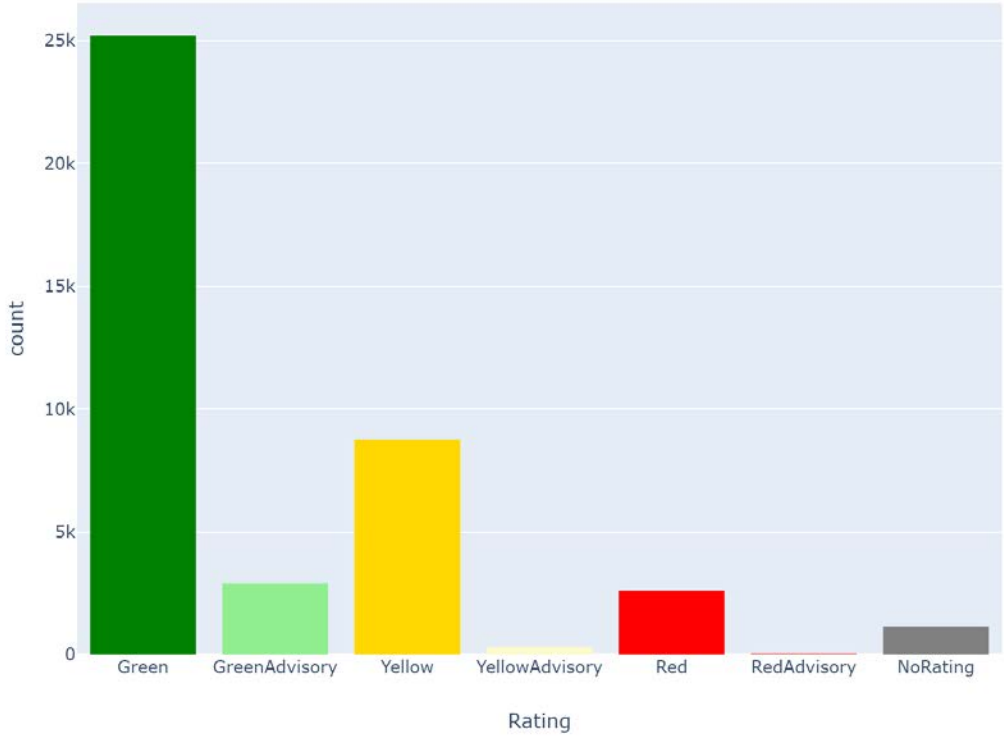

Figure 5. Class distribution of program manager ratings

Since less than 1% of the data is Yellow Advisory or Red Advisory, those rating categories are removed from the data, and the final data set has the following breakdown for a total of 40,485 data points shown in Table 1.

Table 1. Final class breakdown of program manager ratings

Rating	Percentage
Green	62%
Green Advisory	7%
Yellow	22%
Red	6%
No Rating	3%

The cleaned data is separated into test and training sets. 20% of the data is reserved for testing, and the remaining is used to train the model. A vocabulary of words appearing in the data set is then built from the training data. Further text cleaning is done to convert all words to lowercase, remove all nonalphabetic characters (i.e., digits, &, -, :, etc.) and remove all stopwords (i.e., the, at, by, for, etc.). Only words with a frequency greater than two are kept in the dictionary. This helps to eliminate words that are misspelled or only appear in a few of the explanations. The final vocabulary is 26,055 words, which will be used to train the model. We can see the most frequent word and word pairs from the training data set in Figure 6 and Figure 7.

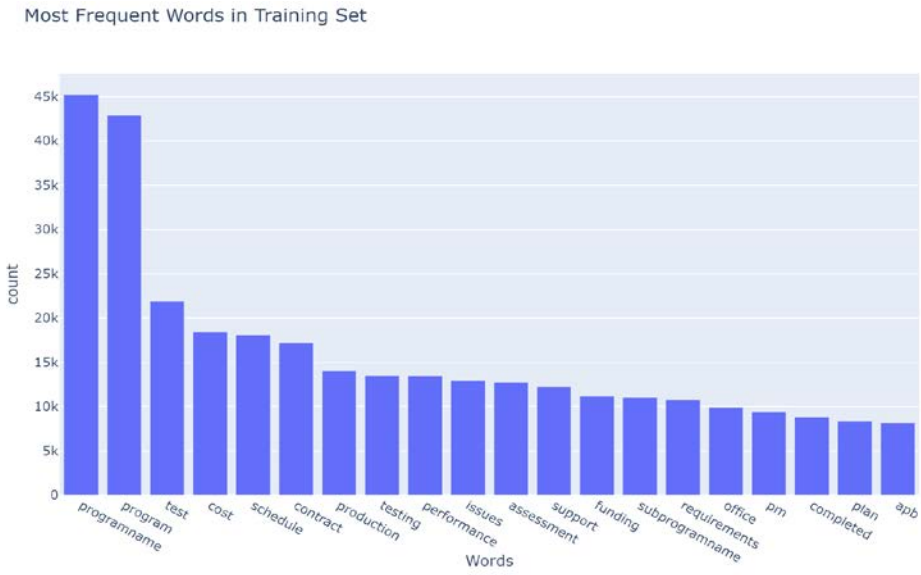

Figure 6. Most frequent words

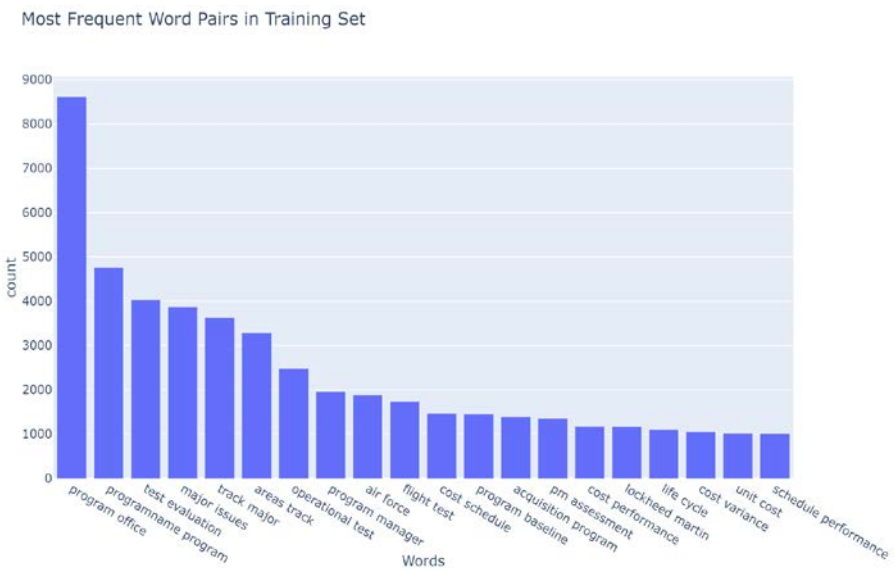

Figure 7. Most frequent word pairs

The most common words align with what we would expect from an acquisition data set. With "test," "cost," and "schedule" being some of the most frequent words and "program office" and "test evaluation" being some of the most frequent word pairs. A word embedding is built based on the words in the training data set. That is, each word is transformed into a numerical representation based on its context, among other words. While there are pre-trained embeddings available for use in natural language processing tasks such as the word2vec embedding, which is built from a corpus of Wikipedia text, for this data set a new embedding was built. The benefit of this is that words or tokens specific to defense acquisition (e.g., RDT&E, MILCON, or APUC) will be encoded with meaning, whereas using a pre-trained embedding would eliminate any unknown words from the text fed to the model. This numerical representation is then fed into a neural network to predict the rating. Several different parameters were tested and the final model had an accuracy of 98% on the training set. The model is then evaluated on an unseen set of test data, and the accuracy is 87%. The accuracy of the model indicates that it performs well on unseen data and is not overfit.

Discussion & Conclusion

The model results show that natural language processing models can have impressive results when applied to acquisition data. The question that remains is how to best apply models such as these in the defense acquisition environment. Generating predicted ratings may not be particularly useful for OSD analysts, but rather than looking through hundreds of ratings and explanations natural language processing can be used to highlight words of interest in an explanation for analysts to view. The LIME (Local Interpretable Model-Agnostic Explanations) model can be used to highlight which words and tokens contribute most to a prediction (Ribeiro, Singh, & Guestrin, 2016). An example of the output of this model with the NLP model as input is shown in Figure 8 below. This figure shows first the prediction probabilities for each available rating; then for each of those ratings, it shows which words either positively (to the right) or negatively (to the left) contribute to the model predicting that rating. The input text is also shown with the words of interest highlighted. Note that the text has stopwords and other symbols removed (as described in the methodology) so it may seem garbled. In the example above, the actual rating from the program manager was yellow (not shown), but the model predicted red 69%

of the time. We can see that the words that positively contributed to the model predicting red include “developing,” “unable,” and “problem.” Similarly, “problem” negatively contributed to the model predicting the rating as yellow (i.e., the word “problem” caused the model to “think” that the text was not rated yellow). Reading the text, we can get a sense that perhaps this is a program that is recovering from an issue and words of interest have been highlighted for us.

Figure 8. Output from the LIME model for a program given a “yellow” rating

Though the model incorrectly predicted the rating of this text, this may be a case where an analyst should look more closely at the program attributes to determine if the rating is correct or if the program needs further investigation. This type of model explainability allows human analysts to be able to understand the decision of the machine. Combining the skills of a human analyst with the power of a machine learning model, we can more quickly assess which programs should be looked into further.

Another area that may be of interest as a result of this work is acquisition policy development. Of the available data, only about 20% was used in the model since the remaining entries either had no explanation, no rating, or the rating or explanation did not contain valuable information (e.g., “Program on track. Not applicable.”). Approximately 4,000 of those unused entries were rated “red” with no explanation. This type of analysis indicates that perhaps acquisition policy could be adjusted to gain more useful insights into the performance of the program. Similarly, some entries had an explanation, but the rating was listed as “no rating.” Upon examining the data, we can see that the text indicates that some sort of rating should be given. In Figure 9 below, a genericized example of this is shown. We can see for the highlighted instance the given rating was “no rating” and the model predicted “yellow.” The text indicates that this perhaps should have been a yellow rating and instead the program manager mistakenly did not enter the appropriate rating. Policy could be enacted that would require program managers to enter only one of a given set of ratings and that “no rating” could not be given, especially if there is an explanation given.

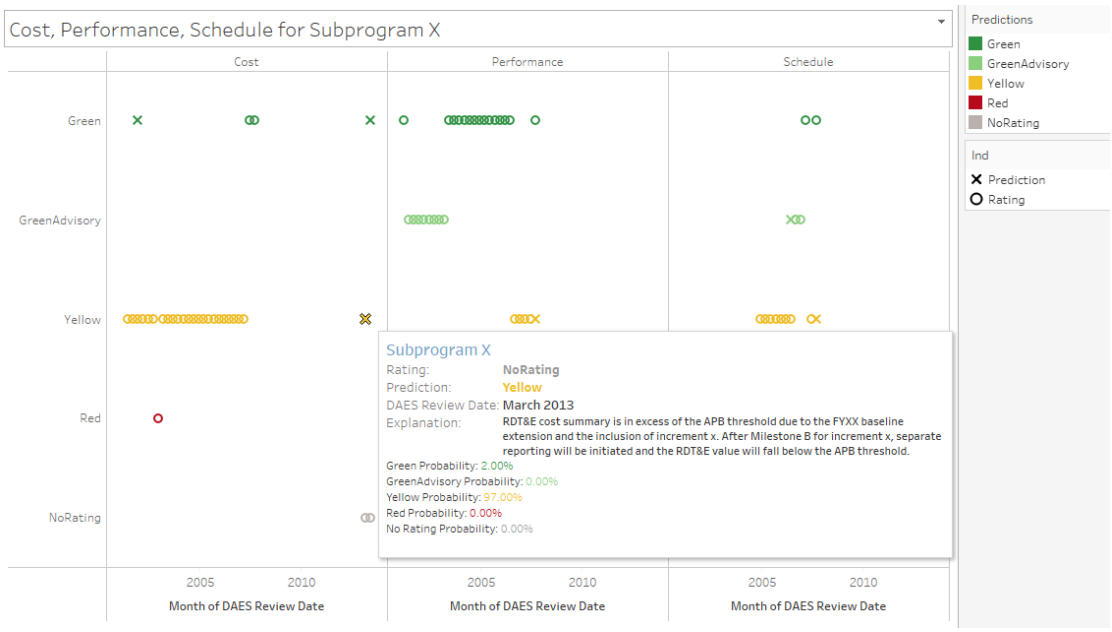

Figure 9. Ratings for a sample subprogram with highlighted inaccurate No Rating given

There are limitations to the model that could be further explored. For example, the dictionary of stopwords that was used eliminated negation words such as “no” and “not” as well as quantity words such as “few” and “more.” Future iterations could better refine the list of stopwords that are removed from the text. Additionally, the text could be further cleaned by either replacing common acronyms with their meaning or vice versa. This would give more consistency to the embeddings so that acronyms such as O&M would always be understood to be synonymous with operations and maintenance rather than having separate numeric representations for each of them. This iteration of the model retained the use of the green advisory and no rating categories, but since advisory ratings are no longer in use and no rating is arguably not useful to predict, these could be coded as a different rating or eliminated from the data set. Machine learning models can be most effective when used in parallel with human analysts as there can be unknown biases encoded into the data and therefore into a model that a machine would not be able to recognize but would be apparent to an analyst. This is why model interpretability through the use of techniques like LIME are useful in implementing machine learning models.

In conclusion, we have shown that natural language processing models can provide a wealth of new information to support decision-makers in defense acquisition. Particularly, a model built from the quarterly DAES assessments has the potential to ease the workload of Department analysts and support them in their use of program ratings to provide portfolio and program insight.

References

Canuma, P. (2019, August 29). The brief history of NLP. *Data Driven Investor* (blog). Retrieved April 8, 2020, from https://medium.com/datadriveninvestor/the-brief-history-of-nlp-c90f331b6ad7

Conneau, A., Schwenk, H., Barrault, L., & Lecun, Y. (2016). Very deep convolutional networks for text classification. Retrieved from https://arxiv.org/abs/1606.01781

Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017). *Language modeling with gated convolutional networks.* Retrieved from https://arxiv.org/pdf/1612.08083.pdf

Garbade, M. J. (2018). A simple introduction to natural language processing. Retrieved April 8, 2020, from https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. Retrieved from http://arxiv.org/abs/1705.03122

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. Retrieved from http://arxiv.org/abs/1301.3781

Mikolov, T., Yih, W.-T., & Zweig, G. (2013). *Linguistic regularities in continuous space word representations.* Association for Computational Linguistics. Retrieved from http://research.microsoft.com/en-

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016).”Why should I trust you?” Explaining the predictions of any classifier. Retrieved from http://arxiv.org/abs/1602.04938

Zaman, M. M. A., & Mishu, S. Z. (2017). Convolutional recurrent neural network for question answering. In *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)* (pp. 1–6). IEEE. Retrieved fromhttps://doi.org/10.1109/EICT.2017.8275236

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net