SYM-AM-22-075

Excerpt from the Proceedings of the Nineteenth Annual Acquisition Research Symposium

Acquisition Research: Creating Synergy for Informed Change

May 11–12, 2022

Published: May 2, 2022

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

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School

The research presented in this report was supported by the Acquisition Research Program 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).

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School

Recommending Recommendations to Support the Defense Acquisition Workforce

Dr. Carlo Lipizzi—is a professor at the Stevens Institute of Technology teaching, researching, and consulting on Machine Learning and Natural Language Processing. He has a PhD in System Engineering from Stevens Institute of Technology, an Executive Management Degree from IMD, Lousanne, CH, and a master's in mathematics from the University of Rome, Italy. As director for the Center for Complex Systems and Enterprises, his research focus is on Machine Learning, Data Science and Natural Language Processing. [clipizzi@stevens.edu]

Hojat Behrooz—works as a data scientist with Stevens Institute of Technology, School of Systems and Enterprises. He has joined Stevens with years of knowledge and experience in intelligent transportation systems. Over the years, he has led several major transportation and traffic operation projects improving the safety, environment, and equity in Tehran's megacity. His efforts to design and implement the first BRT lane was recognized and awarded the Sustainable Transport Award at the TRB conference in Washington, D.C., in 2010. Behrooz has a BSc in Computer Engineering and a PMP certificate from the Project Management Institute. He received his MSc in Engineering Management from Stevens in 2021. His MSc dissertation was on machine learning application in surface transportation systems. [hbehrooz@stevens.edu]

Michael Dressman—is currently a second semester graduate student in the Master of Engineering in Systems Analytics program at Stevens Institute of Technology. He received a Bachelor of Science in Computer Science at Lehigh University with minors in Data Science and Economics. Currently a research assistant in the School of Systems & Enterprises, Michael is interested in data analysis and modeling along with natural language processing and understanding. [mdressma@stevens.edu]

Arya Guddemane Vishwakumar—is currently a second semester graduate student in the Master of Science in Computer Science program at Stevens Institute of Technology. He received a Bachelor of Engineering and Computer Science from the Dayananda Sagar College of Engineering, Bangalore. He has experience in Machine Learning and in front-end development. Currently a research assistant in the School of Systems & Enterprises, developing the user interface for this project.

Kunal Batra—is Manager, Information Systems and Technology for the Systems Engineering Research Center as well as for the School of Systems and Enterprises within Stevens Institute of Technology. He is responsible for the overall planning, organization, and execution of all information technology within the school. He analyzes and manages servers, systems, and assets and recommends and implements solutions. He also carries out support and maintenance of existing applications and development of new technical solutions, leading the developers on software projects serving as a liaison between business and technical aspects of each stage. [kbatra@stevens.edu]

Abstract

This paper presentings the preliminary results of a research study to support the Defense Acquisition Workforce with a Natural Language Processing (NLP)/Machine Learning (ML) prototype of a system to determine what are the most relevant recommendations that stakeholders are providing to the Defense Acquisition community.

The problem addressed by the research study is in the realm of NLP and ML and it is part of the quite popular category of "recommendation systems." Unlike the majority of the cases in this category, though, this task does not focus on numerical data representing behaviors (like in shopping recommendations), but on extracting user-specific relevance from text and "recommending" a document or part of it.

In order to identify important pieces of these texts, subjective text analysis is required to be run. The method used for the analysis is the "room theory framework" by Lipizzi et al. (2021) which applies the Framework Theory by Marvin Minsky (1974) through the use of text vectorization. This framework has three main components: a vectorized corpus representing the knowledge base of the specific domain (the "room"), a set of keywords or phrases defining the specific points of interest

for the recommendation (the “benchmarks”) and the documents to be analyzed. The documents are then vectorized using the “room” and compared to the “benchmarks.” The sentences/paragraphs within a given document that are most similar to the benchmarks, and thus presumably the most important parts of the document, are highlighted. This enables the DAU reviewers to submit a document, run the program, and be able to clearly see what recommendations will be the most useful.

Keywords: Recommendation Systems, Contextual Understanding, Text Mining, Natural Language Processing, Text Vectorization

Introduction

Analyzing text and extracting actionable elements from it is intrinsically challenging, as this task is strongly supported by human common knowledge, and therefore automatic systems fail in true semantic understanding.

Traditional approaches to text analysis are based on “Symbolic” processing, where predefined structures (ontologies and taxonomies) are used to extract semantic elements. The problem with those systems is in the limited context-dependent analysis they can perform, being based on structures that are rarely optimized for the specific need and the given time. This is a “rationalist,” rule-driven approach.

Emerging and new approaches are based on heavy use of Machine Learning and employ complex “deep learning” systems inspired by the human brain structure. The problem wtih those systems is not taking into consideration how humans represent their knowledge and how we achieve the understanding of a problem. This is an “empiricist,” data-driven approach.

Our approach is a combination of Symbolic and Machine Learning, with an additional layer of user interface and visualization, to make the findings more usable by the Defense Acquisition Workforce.

The prototype is based on previous projects we developed for the DoD over the last few years, employing a team of 25 researchers and relying on theories and components we developed for those projects.

For the development of the prototype, we focused on 1) creating a symbolic model for the text understanding and 2) design the process to apply it.

The symbolic model is the “room theory framework” by Lipizzi et al. (2021) which applies the Framework Theory by Marvin Minsky (1974) through the use of text vectorization. This framework has three main components:

- a vectorized corpus representing the knowledge base of the specific domain (the “room”);
- a set of keywords or phrases defining the specific points of interest for the recommendation (the “benchmarks”);
- the documents to be analyzed.

The documents are vectorized using the vectors in the “room” and compared to the “benchmarks.” The sentences/paragraphs within a given document that are most similar to the benchmarks, and thus presumably the most important parts of the document, are highlighted.

The process is a set of logical steps including:

- document ingestion from pdf to text via either the graphical user interface or from existing files;
- text cleaning and "n-gramming" (extracting logical elements composed by multiple words);

- rearranging the document in logical paragraphs;
- vectorize the corpus/knowledge base;
- compare documents to be analyzed with the benchmark elements,
- highlight the most relevant sentences/paragraphs in the original documents;
- present the results via graphical user interface.

Literature Review

Recommendation systems are known commonly to be used to recommend what product you should buy or what movie/show/video you should watch. These systems typically use market basket analysis also known as association rules (Agrawal et al., 1993). Having history on what a specific person or account has consumed helps the systems guess what they would consume next. Items that are consumed together or within a short time frame apart are often considered similar conceptually. If one buys peanut butter, they will likely buy jelly. If someone watches *Star Wars: A New Hope*, then recommending that they watch the sequel *Star Wars: Empire Strikes Back* will more often yield in positive results. Along with products and media, text can also be recommended. While existing techniques for processing text are based on retrieving facts, processing of subjective information is still developing. For subjective analysis Machine Learning is commonly used. Opinion Detection (Jimenez-Marques et al., 2019), sentiment analysis (Pinto & Maurari, 2019), and the use of fuzzy rules to improve text summarization (Guolarte et al., 2019) are all examples. Li et al. (2019) used subjective queries for databases, while Wu et al. (2019) developed an algorithm to account for subjectivity in crowdsourced label aggregation.

Finally, a study from 2006 (Lin et al., 2006) highlighted the need for a perspective analysis when detecting subjectivity in text. This line of study became known as stance detection and is commonly used in opinion mining, to identify if the author is in favor or against the object being analyzed (D'Andrea et al., 2019).

Similarity plays a big role in textual recommendations. With extractive text summarization, text is compared to itself in order to find the sentences or paragraphs that are the most similar to all the other sentences/paragraphs. This technique allows text to be represented by a subset of itself without losing too much meaning. Unfortunately, this technique does not work well with large amounts of text which rules it out of the possibility of being used to recommend parts of lengthy DAU documents.

Existing techniques on textual information processing concentrate on mining and retrieval of factual information (e.g., information retrieval, text classification, text clustering, among others). On the other hand, the processing of subjective perceptions, such as emotions, opinions and summarization, is still a developing field. In particular, because of the intrinsic subjectivity of the summarization process, a generalized summarization model has never been developed.

The automatization of subjective/context dependent tasks is not new in Natural Language Processing. Many efficient algorithms, tools, and techniques have been developed in the past few years and can deliver reasonable results. More recent studies appear to focus on improving these existing methods or creating frameworks that combine them for a certain application.

No one of the above methods, techniques, algorithms could be fully applied to our task. We then opted for an approach—the "room theory framework" by Lipizzi et al. (2021)—which provides a framework to be used to address the needs in our task.

The Room Theory developed by Carlo Lipizzi (2021) is based on Mavin Minsky's Framework Theory (1974). In Minsky's theory, he said that a frame is like a data structure that can express/simplify a concept of being in a room. Lipizzi's theory adds onto this the idea of having a computational version of semantic rooms for Natural Language. A "room" represents the knowledge of a specific domain, it has been created from large corpora related to that domain and transformed into vectors for the analysis.

The main idea is to be able to identify certain structures that would classify the document as belong to a specific domain. In this particular case it is to find recommendations inside recommendation documents. The theory leverages "benchmarks" that are keywords or phrases and finds similarity withing the documents to those benchmarks. The benchmarks are curated by subject matter experts to be able to identify relevant sentences/paragraphs. The overall process of the room theory is displayed in Figure 1 below. Documents are used to train a vectorization model to represent each document as an array of word vectors. Documents can then be checked for similarity with the benchmarks and be classified as important/relevant based on the similarity scores. For this task, we used Word2Vec by Mikolov et al. (2013) as vectorization model.

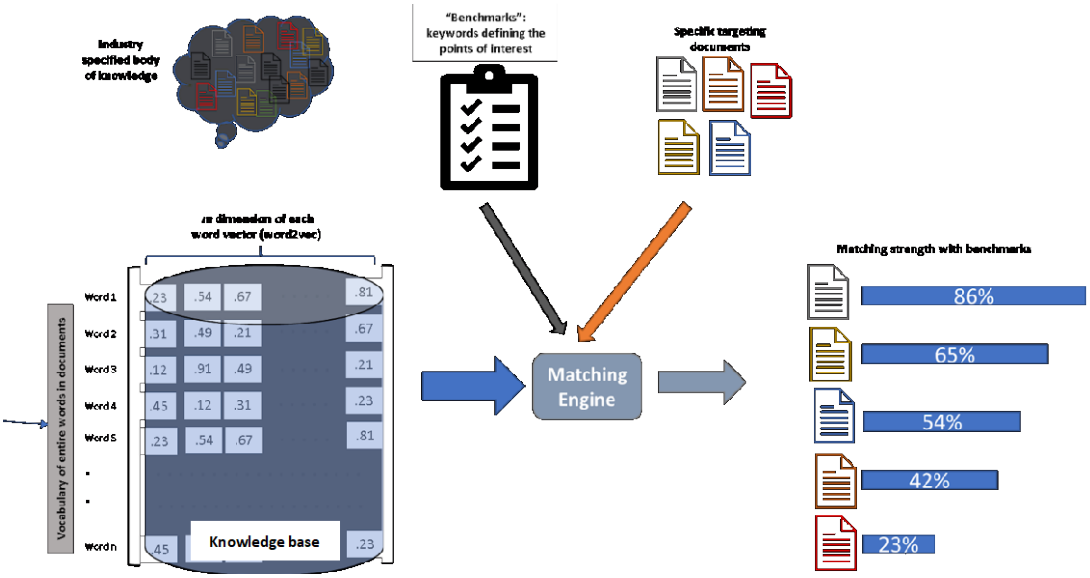

Figure 1. Room Theory Process

Method

To create the room representing the knowledge base for this task, we used a corpus we collected for a previous DAU project. The corpus is composed by1493 pdf files with a total size of 3.1 GB. They were used as domain-specific materials which contained documents related to the DoD and DAU in general. The text was retrieved from these documents using the python library Fitz (PyMuPDF, 2022). Once the pdf files were read in text format, they were then passed to the preprocessing phase. In this phase, the documents were tokenized into words and then preprocessed with steps that include the removal of extra spacing, punctuation, digits, and non-English characters as well as the creation of bigrams and trigrams to be considered as single logical words. After the preprocessing phase, the cleaned word list of each document has been forwarded to a vectorization modeler (Gensim's word2vec) with the following hyperparameters.

```
w2vec_model = Word2Vec(min_count=10,
                       window=7,
                       vector_size=300,
                       sample=6e-5,
                       alpha=0.03,
                       min_alpha=0.0007,
                       negative=20,
                       workers=cores-1)
w2vec_model.train(docs, total_examples=w2vec_model.corpus_count,
                  epochs=50, report_delay=1.0)
```

The model's output contains word embeddings (word vectors) for 27,229 unique words and n-grams after being trained on 7,826,687 total words and n-grams from the entire input documents. The vector size was 300 dimensions. A sample vocabulary word with their frequency is presented in Figure 2 as a word cloud.

Figure 2. Wordcloud For the Trained Model Vocabulary

In order to create the "benchmark," we assembled a list of words defining the elements of interest for recommendation. The list has been developed with subject matter experts. The initial list is filtered and enhanced with synonyms and misspellings, and then a weight value between 1 to 5 is added to the list. This weight shows the importance of the benchmarks for the targeted subject. Each "word" in the benchmark is actually a list of words, with the original word as a root and additional words being synonyms and misspellings, to improve the benchmarking process. For example, for the word *optimization* would have also terms such as optimizing, optimization, optimizations, optimums, optimizes, optimum, optimizations, optimized, optimize, optimally, optimize, and optimal. By applying this technique to 173 initial benchmarks, 1196 total benchmarks and their roots are created as the primary benchmark list; 423 items of this benchmark are not defied in the trained vocabulary and ignored for further processing. Therefore, the final benchmark list consists of 773 known benchmarks by the vocabulary.

Table 1. A Sample of the Benchmarks and Their Weights

Seq	Benchmark	weight	Seq	Benchmark	weight	Seq	Benchmark	weight
1	accurate	1	31	feasibility	2	61	priority	1
2	achievable	1	32	finalize	2	62	programs	1
3	achieve	1	33	framework	1	63	progress	1
4	act	2	34	goals	1	64	quality	1
5	address	2	35	guidance	2	65	rebuild	2
6	advantages	1	36	identified	1	66	recommend	5
7	align	2	37	implement	2	67	recommendation	5
8	analytical	1	38	improve	1	68	replacement	1
9	assess	2	39	incomplete	1	69	require	1
10	assessment	2	40	incorporate	2	70	requirements	1
11	assignment	1	41	integrated	1	71	revise	4
12	baseline	1	42	lack	2	72	revised	4
13	capability	1	43	lifecycle	1	73	risks	1
14	capture	1	44	maintain	1	74	root cause	1
15	challenges	1	45	manage	1	75	schedule	1
16	challenging	1	46	measure	1	76	setting	1
17	completed	1	47	metrics	1	77	should complete	3
18	conduct	1	48	missions	1	78	should follow	3
19	configured	1	49	modernize	2	79	strategy	1
20	construct	1	50	monitor	1	80	structure	1
21	coordinate	1	51	monitoring	1	81	sufficient	1
22	costs	1	52	moving forward	1	82	support	1
23	critical	1	53	needed	1	83	sustainment	1
24	define	1	54	operational	1	84	system	1
25	develop	1	55	optimization	1	85	take action	3
26	development	1	56	performance	1	86	transition	1
27	effort	1	57	plans	1	87	update	1
28	emphasizes	1	58	policy	1	88	weaknesses	1
29	evaluation	1	59	practices	1			
30	execute	1	60	prioritize	2			

Together with the benchmark list and its weights, the trained model provides the essential tools to evaluate the input documents.

The model can analyze a new input document and measure its specificity from several angles. Two more relevant evaluation process is presented here, which are more applicable to the recommendation system.

Relevant/Irrelevant Input Document

This concept gives a measure related to each input document which shows how much the input text file is relevant to the benchmark. In other words, this measure shows the similarity of the content of the input document to the benchmarks in total. This measure would be a value between 0 and 1, in which higher values show more similarity. A threshold should be assumed to separate relevant/irrelevant input documents regarding the benchmarks. In this process, we

used the cosine similarity to create a matrix of similarity between the benchmarks and entire input document words, as shown in Figure 3. Each matrix column represents the whole input document's similarity to a specific benchmark. The frequency distribution of this vector shows how input document words are related to that specific benchmark. A skewed distribution shows that more words in input documents have similarities to the particular benchmark and vice versa.

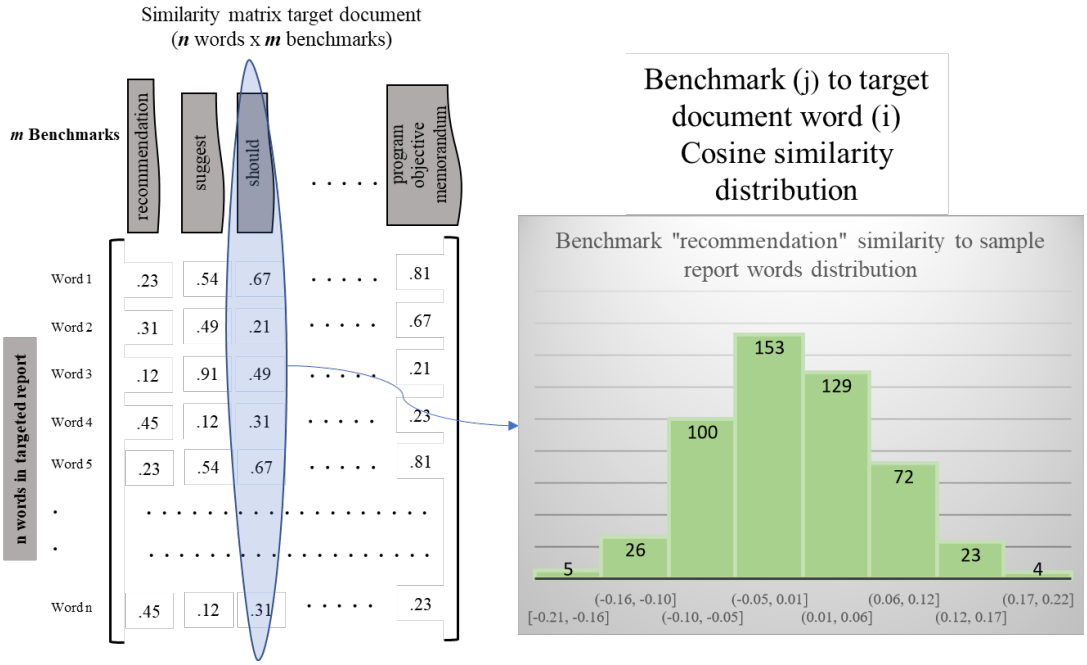

Figure 3. Frequency Distribution of Document's Words and Specific Benchmark's Similarity

A two-bin distribution could be applied((-1,0) and (0,1)). The first bin shows how dissimilar a specific benchmark is to the entire input document. Bin 2 also shows how similar the input document is to the specific benchmark. This second bin could be used to measure the similarity between the benchmark and the entire input document. All the values from the comparison of benchmarks and the input document words create a similarity vector between the benchmarks and the input document, as represented in Figure 4. This will be used to highlight the document/parts of document that are more relevant.

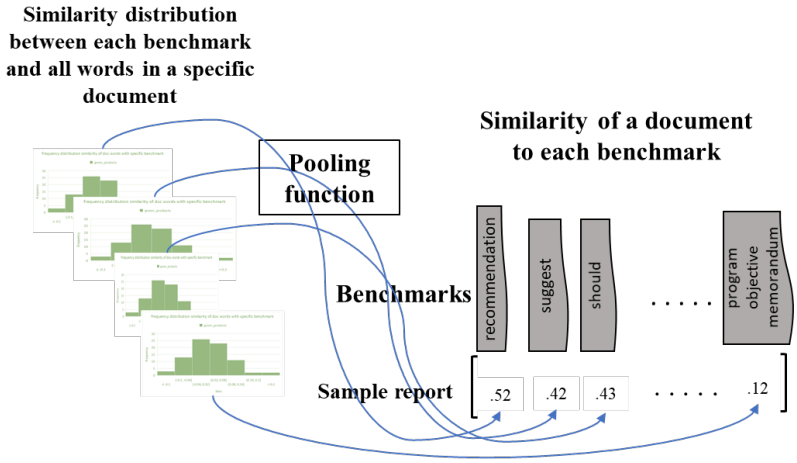

Figure 4. Bin 2 Pooling to Calculate Each Benchmark and Entire Document Words

Another element that can be extracted is drilling down the relevance of each word in the benchmark for each document or part of it. A sample result of the small set of benchmarks and input documents is presented in Figure 5.

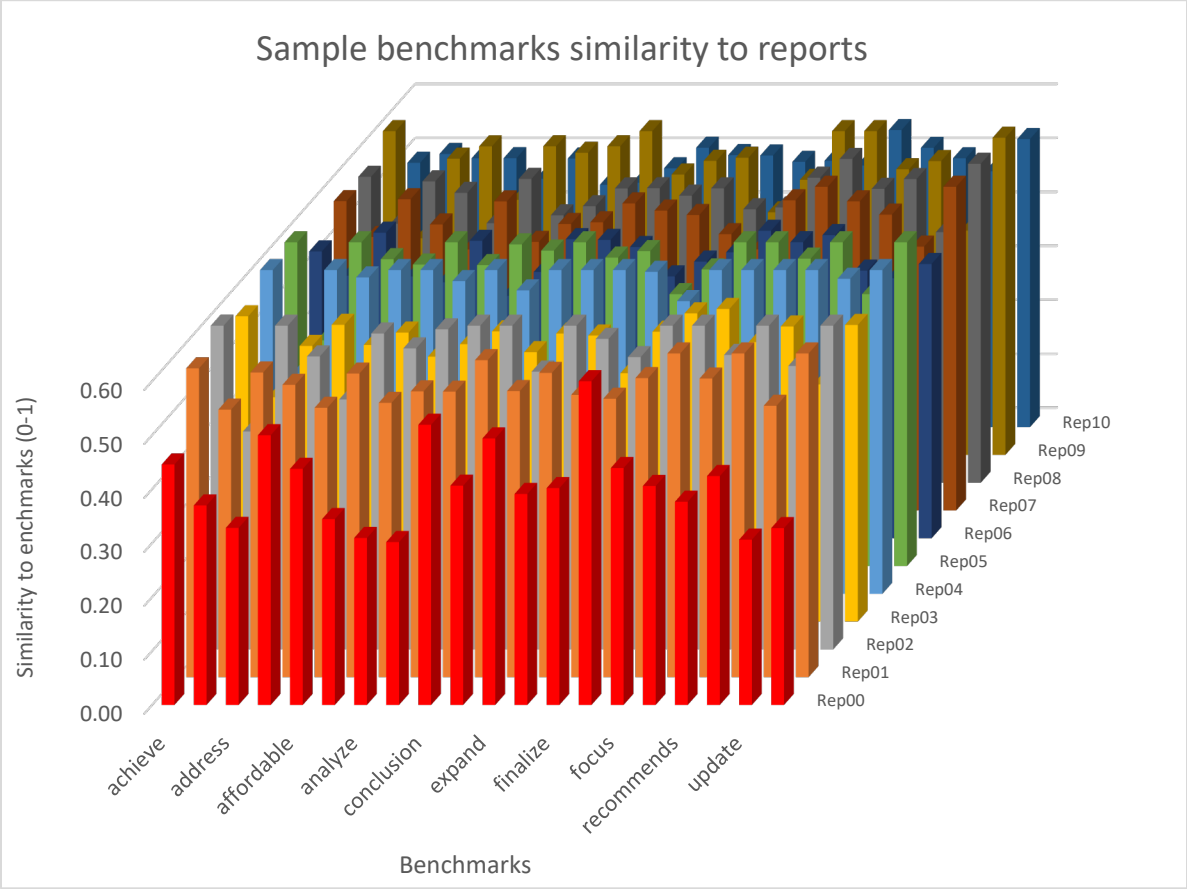

Figure 5. The Similarity of Entire Words of Documents and Each Benchmark's Comparison

As it can be seen, the Rep00, in general, is less similar to the sample benchmarks, while the Rep04 has more similar measures to the benchmarks. By counting the number of words with a similarity more than a threshold (such as 0.50) and normalizing it with respect to the number of the words in each document, an average similarity measure is calculated that presents the level of similarity of the entire document with respect to the entire benchmarks. This single measure for each document can be used to compare various documents to each other in similarity to entire benchmarks. A document with a higher measure is more relevant or like the benchmarks. Figure 6 presents a comparison presentation together with an assumed threshold (0.60) for finding relevant/irrelevant input documents. This would provide an overall view of the documents in terms of their relevance.

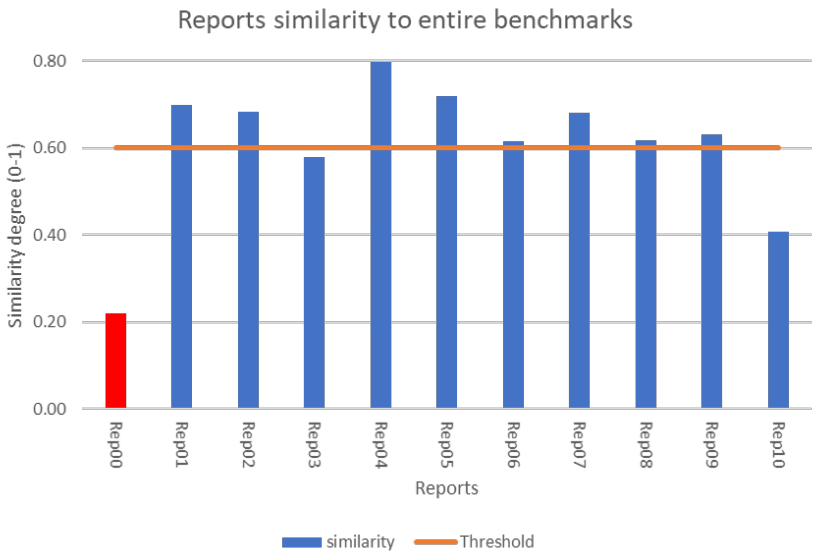

Figure 6. Comparing Various Documents' Total Similarity to Entire Benchmarks

Highlighting the Recommendation Part of an Input Document

As mentioned, we created a similarity matrix for each word in the document and benchmark. The similarity between word n and benchmark m will be from -1 (most dissimilar) to 1 (most similar). The max-pooling technique is then implemented so that there will be an array of length n that will have the maximum similarity between a specific word with the set of benchmarks. Max-pooling is sample-based discretization process to down-sample an input representation by reducing its dimensionality. This is portrayed below in Figure 7.

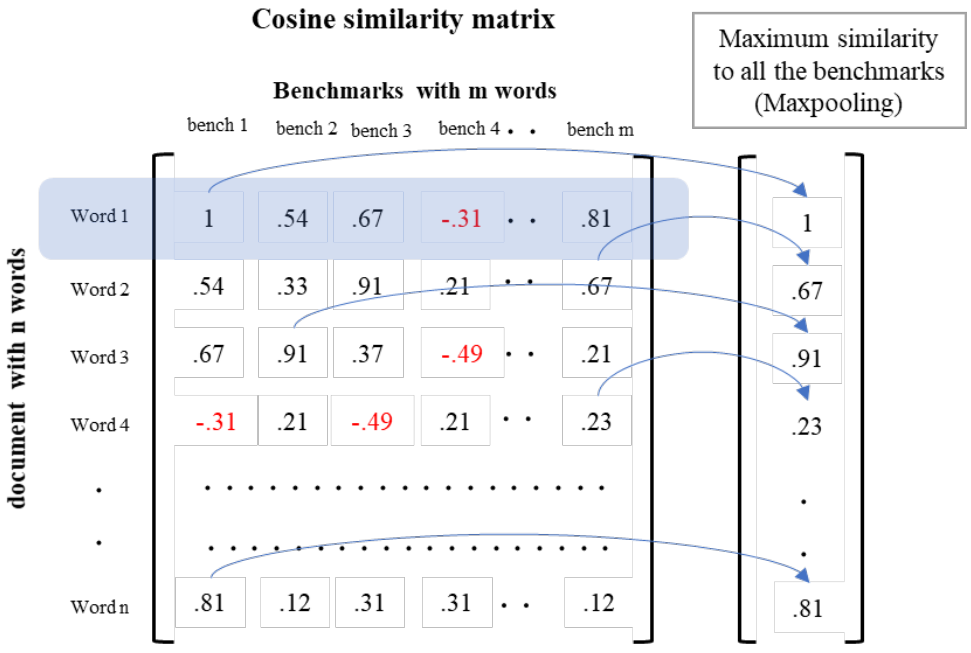

Figure 7. Cosine Similarity Between Document's Word and Various Benchmarks

This max pooling is then weighted, so the max similarity of the word and benchmark is multiplied by the benchmark's weight. So, for example, for word 1, since the maximum similarity is with benchmark 1, the first element of the array would be multiplied by benchmark 1's weight.

To determine the relevant parts of each recommendation, the document was looked at in segments of words. Since words form sentences and sentences form paragraphs when you have relevant words that could be useful for recommendations, the sentences and paragraphs of those words become essential. To account for this, each document's max similarity array had its moving average with a window size of 20 words calculated at each word. In Figure 8, the location of the window of 20 words that maximizes the moving average is shown on a plot of the weighted max pooled similarity over the time of the whole document.

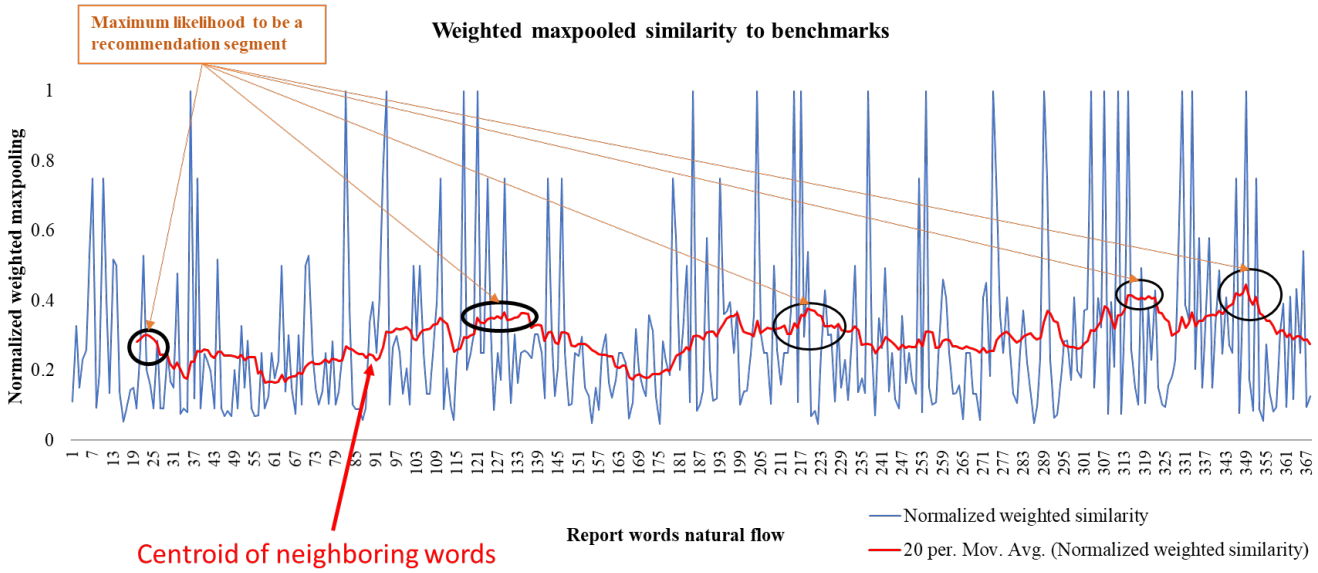

Figure 8. Document Moving Average Similarity to Entire Benchmarks

It was found that with a window of 20 words from the similarity matrix, the actual document (which includes the raw text) would have a window of 35 words that would make up important and relevant recommendations. To assure high-quality moving average windows, the threshold of average similarity is set to 0.75. Any window of words above that threshold is then traced back to the original document and is highlighted. We used PyMuPDF from the Fritz library for this task. An example of what the highlighted section looks like is in Figure 9.

Key Recommendations to Address These Issues:

- The VCSA should co-chair the ASARC with the ASA(ALT); ASARC will make appropriate recommendations to the AAE.
- Capability Portfolio Reviews:
  - The VCSA and the ASA(ALT) should co-chair Session 1 of the materiel CPRs.
  - Codify the conduct of CPRs in an Army Regulation.
  - Include a requirement to review the interdependencies across portfolios.
- Synchronize the ASTAG and ASTWG cycle with the POM submission cycle.
- Improve the alignment among the PEO structure, Equipping PEG, BOSs, CPRs and TRADOC Centers of Excellence.
- Rebuild the highly efficient and effective triad of the military DASC, SSO and PA&E Action Officer (AO) at the O-4/O-5 level, with 'knowledge authority' and locate in the Pentagon.
- Make PMs lead/accountable for acquisition logistics during development through successful IOC fielding and LCMCs lead/accountable for post-fielding operational logistics.
- Disestablish RDECOM and return the RDECs to the LCMC Commanders.
- Establish a MG or SES 5 Executive Director for RDA reporting directly to the CG AMC.
- CMO should promulgate policy and develop metrics for line and staff accountability in Army acquisition.
- Army leadership must improve communication with industry.

Figure 9: Highlighted Text in the Original Input File as a Relevant Part

User Interface

In order to make the system easy to be used by the Defense Acquisition Workforce, we developed a web-based graphical user interface. The interface gets the data from a repository where the documents would be placed and were the temporary results will be hosted. We use a MongoDB database for the repository. The webpage with the user interface contains two separate groups, one for a user and one for an admin. The user logs in with an email and password and they will land at a drop box page shown in Figure 10.

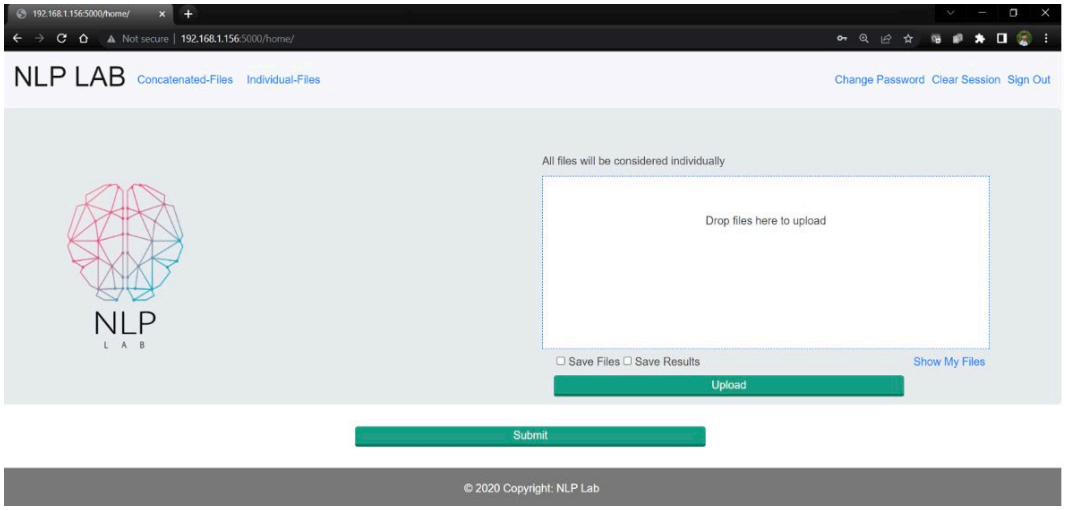

Figure 10. User Homepage

In this drop-box, the user can upload files to the repository for storage. Once uploaded, the file can also be submitted, which would trigger the file being used to run in the recommendation system to get the document similarity to the benchmarks as well as having the document's sections with a moving average similarity over the threshold being highlighted and shown to the user. The user has the option to submit individual files for running through the system or to submit several files that will be concatenated run through the system at the same time.

The admin has the ability to create new users and to manage the users files. Which entails managing the repository on the MongoDB database. In the database, the files are stored in small chunks, as represented in Figure 11.

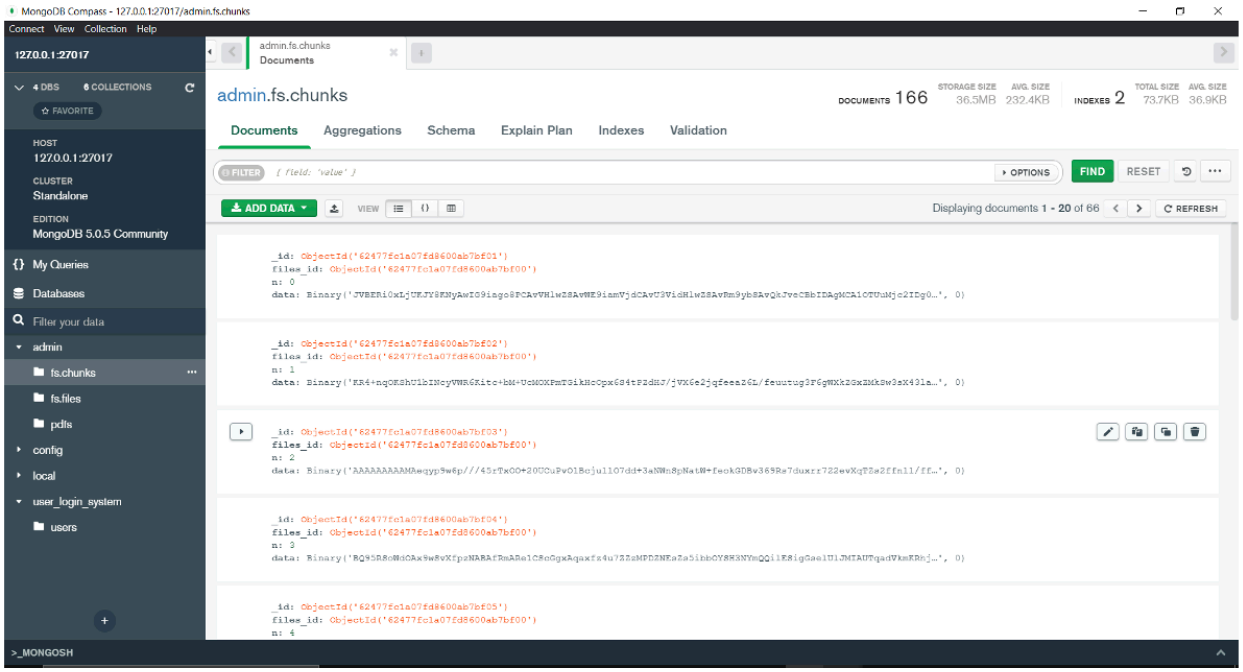

Figure 11. MongoDB Database for File Chunks.

Results

From using 1493 pdf documents and 773 benchmark keywords and phrases in training, 11 documents of varying length were used to evaluate the model. Ten of these documents are recommendation documents, while there is one control document, CMH_Pub_72-2.pdf. Overall, the recommendation documents came back with high similarity regarding the domain-specific benchmarks. A good indicator that the model learned is that the control document's similarity (0.25) was significantly lower than the worst recommendation document (0.5). This means that the model did an accurate job of learning the domain of recommendations and finding the parallels in the documents.

Table 2. Document Similarities Measure to Benchmarks

Input Document	Similarity Degree
AIA - Acquisition Rebalancing 2May2014.pdf	0.80
APMP - Closing the Proc Gap Survey_v6 2014.pdf	0.78
Birkler et al. 2010 - Marginal Adjustments to Meaningful Change - Rethinking Acq RAND_MG1020.pdf	0.66
CMH_Pub_72-2.pdf	0.25
Decker-Wagner Army Acquisition Review - Summary and Implementation 2010.pdf	0.82
GAO - Defense Acquisitions - Where Should Reform Aim Next 29Oct2013.pdf	0.81
Goldwater Nichols - Perfect Storm - Nemfakos Blickstein 2010 RAND OP-308.pdf	0.70
NDIA - Pathway to Transformation Acquisition Report 14Nov2014.pdf	0.78
PSC - Acquisition and Technology Policy Agenda - 28July2014.pdf	0.76
Schmidt 2000 - Acq Reform in US Army - Changing Bureaucratic Behavior - RAND MR-1094-A.pdf	0.70
Sec809Panel_Vol2-Report_June18.pdf	0.50

When the documents' similarity is plotted against the page length in Figure 5, there appears to be a negative correlation. The longer the document is, the lower the similarity score. One exception is the second-longest document, Decker-Wagner, which was 246 pages long and scored the highest similarity. Since there are only 10 data points, it is hard to generalize this to every document, so to understand better if this trend is common, more documents would need to be evaluated by the model. We are also implementing a paragraph-level analysis, as detailed in the conclusions/future development paragraphs. This would provide a better level of granularity in the recommendation: in a longer document there may be parts that are highly relevant, along with others—eventually many others—that are not. This would make the whole document relatively low in relevance, losing the relevance of its key parts.

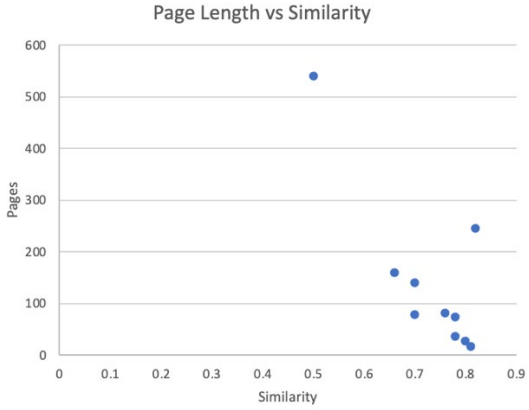

Figure 12. Page Length vs. Similarity

For individual documents, the sentences/paragraphs with high moving average similarities can be highlighted to give the review an easy way to locate the relevant and vital parts of the text. Looking back to Figure 9, the model does a good job of highlighting specific recommendations to be implemented, but the accuracy needs to be improved with more documents to train and tune up the model, addressing in particular larger documents.

Conclusions

This work used the Room Theory created by Lipizzi et al. (2021) to frame the solution to identifying relevant and important documents of varying lengths and specific parts of documents. These documents are specific to the domain of recommendations for the DAU of the DoD to implement. This approach used 3.1 GB of documents to train a vectorization model to create a vocabulary of 300-dimension word vectors. The documents were compared for similarity to the benchmark keywords and phrases on a word-to-word level. Moving average similarities were calculated to highlight the relevant/important parts of the documents for review without skimming the whole text. For evaluation, we used 10 recommendation documents and one control document, where the 10 documents scored relatively high while the control scored poorly as expected.

For future improvement, the documents could be broken up into sentences as a whole document doesn't always have the same central point. Once broken up, the sentences would be clustered together by similarity to form more cohesive content, creating logical paragraphs. These clusters/paragraphs will then be used as documents in this room theory implementation. This will help combat the issue of low-scoring similarities for lengthy documents. We already developed a prototype for this implementation.

References

D'Andrea, E., Ducange, P., Bechini, A., Renda, A., & Marcelloni, F. (2019). Monitoring the public opinion about the vaccination topic from tweets analysis. *Expert Systems with Applications, 116*, 209–226.

Gourlarte, B. F., Nassar, S. M., Fileto, R., & Saggion, H. (2019). A text summarization method based on fuzzy rules and applicable to automated assessment. *Expert Systems With Applications, 115*, 264–275.

Jimenez-Marques, J., Gonzalez-Carrasco, I., Lopez-Cuadrado, J., & Ruiz-Mezcua, B. (2019). Towards a big data framework for analyzing social media content. *International Journal of Information Management, 44*, 1–12.

Li , Y., Feng, A., Li, J., Mumick, S., Halevy, A., Li, V., & Tan, W.-C. (2019). Subjective Databases. *Proceedings of the VLDB Endowment*, *12*(11), 1330–1343.

Lin, W.-H., Wilson, T., Wiebe, J., & Hauptmann, A. (2006, June). Which side are you on? Identifying perspectives at the document and sentence levels. *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, 109–116.

Lipizzi, C., Borrelli, D., Capela, F. de O. (2021, November 20). *A computational model implementing subjectivity with the "Room Theory". The case of detecting emotion from text* (arXiv:200506059). http://arxiv.org/abs/2005.06059

Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013, September 6). *Efficient estimation of word representations in vector space* (arXiv:13013781). http://arxiv.org/abs/1301.3781

Minsky, M. (1974). *A framework for representing knowledge*.

Pinto, J. P., & Murari, V. (2019, April). Real time sentiment analysis of political twitter data using machine learning approach. *International Research Journal of Engineering and Technology, 6*(4), 4124.

PyMuPDF. (n.d.). PyPI. https://pypi.org/project/PyMuPDF/

Wu, M., Li, Q., Wang, S., & Hou, J. (2019, August). A subjectivity-aware algorithm for label aggregation in crowdsourcing. *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, 373–378.

Acquisition Research Program
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net