The Simplification of Complex Systems using Natural Language Processing

Jaden Flint, Chanler Cantor, William Marx, Ph.D.,
CAPT Timothy Hill, USN (ret.), COL John Frasier (ret.),
Kyle Russell
Intuitive Research and Technology Corporation
Internal Research and Development
Huntsville, AL
jaden.flint@irtc-hq.com; chanler.cantor@irtc-hq.com;
william.marx@irtc-hq.com; timothy.hill@irtc-hq.com;
john.frasier@irtc-hq.com; kyle.russell@irtc-hq.com

ABSTRACT

Increased system capability and heavier dependance on software has resulted in systems that are more complex than ever before. This complexity expansion impacts Systems Engineering analyses throughout the lifecycle of these systems, making system architecting more time consuming. Professional experience and exposure can play a role in the amount of time needed to understand and properly analyze a system, yet increased complexity also increases potential for human error in evaluation and interpretation. Failure to accurately capture system aspects can lead to system failure or, at the very least, the accumulation of technical debt. The farther along the system is in its lifecycle, the more difficult it is to correct the issue. Therefore, the success of the system is dependent on the thorough understanding of the various components comprising the system architecture. The authors present a literature survey highlighting challenges associated with systems architecting and Systems Engineering for complex software systems-of-systems.

The paper presents the authors' research and progress on an innovative application of Natural Language Processing (NLP) to aid the systems engineer, both in terms of comprehensiveness and effectiveness. We are applying NLP to benefit Systems Engineers, specifically those working in the Model Based Systems Engineering (MBSE) domain. The authors evaluate common NLP techniques that can be applied to the highly technical and systematic written language methods used with MBSE. Our paper summarizes contributions to the technical literature in this innovative application of NLP for MBSE. We present a use case application study of the implementation of NLP to provide traceability between a proposed product architecture and a standard for compliance.

ABOUT THE AUTHORS

Ms. Jaden Flint is a Model Based Systems Engineer at Intuitive Research and Technology Corporation. She is responsible for providing technical and programmatic consultation to programs in support of the Modular Open Systems Architecture (MOSA) Transformation Office (TO). She and her team apply knowledge of Systems Engineering methodology to develop model representations of the system using modeling languages such SysML and UAF. She then evaluates the status, effectiveness, and efficiency of existing and future programs on their conformance to model guidance to ensure the MOSA objectives are met. She also manages project schedules and collaborates with peers to ensure that project requirements meet customer expectations and ensures timely deliverables. She has obtained both Level 1 and Level 2 Fundamental OSCMP Certifications. She received her BS in Electrical Engineering from The University of Alabama in Huntsville and is currently pursuing her Master's in Business Administration.

Ms. Chanler Cantor is an Area Leader at Intuitive Research and Technology Corporation. She is responsible for managing the Internal Research and Technology portfolio and leads multiple projects associated with Artificial Intelligence and Machine Learning, Big Data Analytics, complex visualization, and medical imagery visualization. She and her team develop technology demonstrators and prototypes which can be showcased to customers to promote awareness and early evaluation and adoption of technologies. She has been awarded six patents related to big data visualization and other emerging technologies. She received her MS in Systems Engineering from Johns Hopkins University and received her BS in Electrical Engineering from The University of Alabama.

Dr. William Marx is the Senior Vice President and Chief Technology Officer of Intuitive Research and Technology Corporation in Huntsville, Alabama. He is responsible for planning, managing, and executing research and development programs aligned with the technology priorities of the U.S. military and commercial customers. His experience base and technical portfolio include advanced visualization systems, Big Data analytics, Artificial Intelligence and Machine Learning, Knowledge Based Systems, missile system design, multi-disciplinary design optimization, missile guidance and control, analysis of exo-atmospheric kill vehicles, supersonic aircraft design, and ground and space robotic system design. Dr. Marx received his PhD and MS in Aerospace Engineering from the Georgia Institute of Technology and his BS in Aerospace Engineering with a minor in Mathematics from Embry-Riddle Aeronautical University. He was a NASA Langley Graduate Student Researchers Program (GSRP) Fellow.

CAPT Tim Hill, USN, (ret) is the Director of Central Florida Operations with Intuitive Research and Technology Corporation (*INTUITIVE*). He is responsible for efficient operation of the Central Florida office and for representing *INTUITIVE* in Central Florida with all customer sets and industry and academic teammates. He is a retired Navy Officer who served across a wide range of operational, staff, and acquisition assignments, culminating in his final tour commanding the Naval Air Warfare Center Training Systems Division (NAWCTSD). He amassed over 3,200 flying hours and 700 carrier arrested landings in more than 32 aircraft types, including operational assignments in the S-3B Viking and F/A-18F Super Hornet. He earned a BS in Systems Engineering from the U.S. Naval Academy, a MS in Systems Engineering from Johns Hopkins University, and a MS in International Relations from Troy University. He is a graduate of the U.S. Naval Test Pilot School and the Air Command and Staff College. He has published articles and papers on topics ranging from tactical aircraft operations to technical studies and acquisition best practices. He is an active member of the Central Florida community through board service and other similar activities.

COL John Frasier, USA, (ret) is the Director of Aviation Operations at Intuitive Research and Technology Corporation where he provides leadership, subject matter expertise, and strategic planning for all Aviation related capabilities and pursuits. Previously, he served over 26 years in various command and leadership positions as an Army Aviator, Instructor Pilot, Experimental Pilot, and Acquisition Officer, culminating his career as the Project Manager for Aviation Mission Systems and Architecture (PM AMSA). In this role he was responsible for the execution of the acquisition life-cycle functions of 53 programs that supported all Army Aviation Platforms world-wide, to include the development of the Aviation Mission Common Server (AMCS), Degraded Visual Environment (DVE), Assured Position Navigation and Timing (APNT) capabilities, Network and Communications capabilities, Mission Command capabilities, Air Traffic Control, and Aviation Ground Support Equipment. John holds a MS in Aerospace Engineering from the Georgia Institute of Technology, a Master of Strategic Studies from the United States Army War College, and a BS in Mechanical Engineering from the United States Military Academy.

Mr. Kyle Russell is a Senior Digital Engineer with Intuitive Research and Technology Corporation. As a member of the research and development team he is responsible for exploring new applications of cutting-edge technologies to customer problems. He received a BS in Electrical Engineering from the University of Alabama and is currently pursuing an MS in Computer Science with focuses on Artificial Intelligence/Machine Learning and Data Analytics. He has experience developing advanced real-time interactive visualization solutions, and Big Data Analytics pipelines, as well as experience with modern web technologies and database systems in general.

The Simplification of Complex Systems using Natural Language Processing

Jaden Flint, Chanler Cantor, William Marx, Ph.D.,
CAPT Timothy Hill, USN (ret.), COL John Frasier (ret.),
Kyle Russell
Intuitive Research and Technology Corporation
Internal Research and Development
Huntsville, AL
jaden.flint@irtc-hq.com; chanler.cantor@irtc-hq.com;
william.marx@irtc-hq.com; timothy.hill@irtc-hq.com;
john.frasier@irtc-hq.com; kyle.russell@irtc-hq.com

INTRODUCTION

Model Based Systems Engineering (MBSE) is a new technique developed for performing Systems Engineering using a model centric approach. MBSE provides a digital representation of a system using models which is a very different approach when compared to the traditional document-based approach to Systems Engineering. This technique was developed to extend the effectiveness of Systems Engineering by bridging the gap in communication between diverse audiences. As systems progressively become more complex, the traditional document- based Systems Engineering approach begins to fall short in its ability to reach various stakeholders. With the ever-increasing emphasis of system of systems interoperability, the representation of information becomes more of a challenge. This is because systems are not the only factors increasing in complexity. Language is also expanding. With more verbosity in describing the aspects of a system, the result is an increase in the length of documentation. This can lead to many complications that identify weaknesses within the document-based approach. Change mitigation becomes a challenge because modifications must be tracked and adjudicated in every existing instance of the documentation. With documentation length increasing, there are proportionally more instances to track, further exacerbating existing security threats with even more information to capture. The introduction of models presents a mitigation to these challenges.

Models provide a visual representation of a system that promotes collaboration and understanding of complex systems amongst diverse development teams. All proponents of a system, such as requirements, mission scenarios, business objectives, etc., can all be captured in the model and generated in a view that encourages engagement. With all associated documentation in one centralized location, this alleviates the paper trail and mitigates the security threat that magnifies the complexity of a system. MBSE models introduce the capability to interact with various software applications like Doors, Datahub, etc. These tools can be used to expand the collaboration space and result in more efficient systems overall. Digital models also allow efficiency in recognizing interoperability requirements and mitigating those changes.

The usage of models with Systems Engineering introduces a visual aspect of data representation. This technique expands the capability of systems engineers as it enhances their ability to effectively communicate key factors of system development and progression, such as goals, expectations, milestones, etc., to stakeholders. Through this improved line of communication, errors are more easily identifiable as stakeholders can now cut through the complexity of systems to better analyze the details. This allows the identification to occur in earlier stages of the systems lifecycle. Correcting said errors is vital to the successful simulation of systems and will impact the Department of Defense training community by alleviating technical debt at later stages of progression. As errors are identified and adjudicated, the analysis of these complex systems is improved, and simulations are better informed for future developments. Efficient Systems Engineering results in better development and fielding of training systems.

However, systems are continuing to grow in complexity. MBSE models allow a visual aspect, but the Systems Engineering methodology that goes into informing those models is still a very vital component. Without the proper expertise and system knowledge, important information can be overlooked. Having the right Subject Matter Experts (SMEs) involved in the development process is key in constructing an accurate representation of a system. With the diversity in taxonomy and vocabulary there is still the chance for miscommunication which could result in an error in the system model, not to mention the increased risk of overlooking concepts in translation across the system of systems boundary line. Natural Language Processing (NLP) can be used to further improve the MBSE capability. This technique uses the ability to train an algorithm to recognize and correlate language between aspects of a system.

This paper addresses the problem of extensive verbiage and documentation in the Systems Engineering domain by proposing the use of NLP to augment requirements mapping to decrease evaluation time, augment limited staffing, and identify areas of improvement in the documentation.

PRIOR RESEARCH

Systems Engineering Overview

The practice of Systems Engineering can be traced back to work at Bell Telephone Laboratories in the 1940s, where the term Systems Engineering was first coined, and the first Systems Engineering classes were taught. In the 80 years that have passed since the pioneering days of Systems Engineering, the discipline, methods, tools, and practice of Systems Engineering has evolved substantially. Systems Engineering has grown into a multidisciplinary field of engineering that includes the design, development, integration, test, evaluation, management, operations, and sustainment of complex systems over their entire lifecycle. As other disciplines, methods, and tools evolve, they have become interwoven with Systems Engineering methods and tools. The term Digital Engineering has been used for the past several years to encompass the overarching framework and philosophy that guides design and engineering efforts for today's complex systems of systems. Model-based techniques, digital practices, and cloud computing infrastructure are enabling technical innovations that can benefit systems across their physical and digital lifecycles.

The Systems Engineering Vision

The International Council on Systems Engineering (INCOSE) (2022) released "Systems Engineering Vision, 2035, Engineering Solutions for a Better World," in January 2022. The document provides a conceptualized view of how the practice of Systems Engineering will continue to evolve. It is anticipated that System Engineering will heavily leverage our nation's ongoing digital transformation, embracing model-based methods within an integrated digital engineering environment. Control of the digital thread, integrated tool chains and workflows, and Artificial Intelligence (AI) methods will enable stakeholders to seamlessly participate and collaborate across the digital fabric that connects a system throughout its lifecycle. An important element of Systems Engineering, now and with the future vision, is Requirements Engineering – including creation, quality, management, verification, and validation. As systems and systems of systems continue to become more complex and increasingly digital, the challenges associated with Requirements Engineering also proliferate.

Inclusion of AI and NLP in Systems Engineering Workflows

NLP is the sub-discipline of AI that allows computer algorithms to understand text and spoken words. NLP is used to comprehend the written language, including contextual nuances, styles, patterns, similarities, and semantics. This relationship is visualized in Figure 1. Traditionally, this has been achieved by hand-crafted algorithms painstakingly developed for this purpose. These algorithms are based on mountains of human research on the structure and meaning of language and the complex set of rules that coincide. The performance of some of these algorithms has been quite impressive, with chatbots such as ELIZA having existed since 1966, being able to fool at least a few into believing they are human.

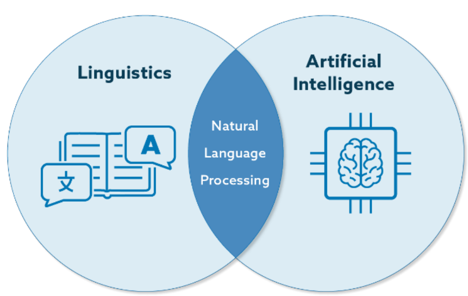

Figure 1. NLP: the Crossover Between AI and Linguistics

Machine Learning (ML) on the other hand is an area of research that uses advanced statistical methods and mountains of examples to "teach" a computer to make some decision based on what it observes in the provided information. The decision does not have to be related to NLP; in fact, it can be anything with the ability to be represented mathematically. One may observe then that the goals of NLP overlap in a significant way with the more general goals of ML and can thus benefit from the research being done in the more general field. And benefit it has, with recent innovations such as Large Language Models having human-like understanding of language.

Researchers are applying NLP to benefit the Systems Engineering practice and methods. As system complexity increases, so do the associated documents, specifications, and models. The design of complex systems is a long, labor-intensive process that relies heavily on domain and Systems Engineering SMEs to shepherd the design elements throughout the development process.

Zhong et al. (2022) have developed an approach to assist systems engineers with the automatic generation of systems diagrams from unstructured natural language text using NLP practices. In this approach, entities and relationships are automatically extracted from specifications, manuals, and technical and maintenance reports; and then are converted into SysML diagrams.

Ambiguity detection in requirements is another challenging problem in the Requirements Engineering community. Bajceta et al. (2022) evaluated several NLP solutions across a common dataset of 180 system requirements. Interpretation of these results is complicated – as is the case with many technology solutions applied to engineering problems, there is no single best answer. Algorithms, models, and learning can be tuned to achieve increased objective performance for some metrics, but often with a decrease in performance for others. Nonetheless, the application of NLP methods within the Systems Engineering domain can contribute greatly to INCOSE's 2035 Vision.

Fuentes (2019) discusses applying AI and NLP to increase the quality of metrics to a level beyond traditional measures of quality. Systems Engineering classes teach proper requirements creation and management, ensuring that requirements are correct, unambiguous, complete, consistent, verifiable, traceable, etc. Fuentes discusses an approach that includes analysis, use of an ontology, analysis of requirements in terms of pattern and semantics. AI and NLP are applied to reduce the time-intensive evaluation of requirements currently performed in the domain by SMEs.

MBSE methods and tools have also advanced significantly in the past 10 years. The benefits of model-based engineering within a Digital Engineering construct will evolve to fulfill the INCOSE 2035 Vision. Platform and tool developers have created Digital Engineering suites of interconnected models to manage systems throughout their lifecycles. Digital twins are created and used within MBSE environments for trade-off analysis, verification, and validation. Madni (2021) describes a MBSE testbed designed to organize and manage digital artifacts, support experimentation with models, algorithms, and data. Ballard et al. (2020) describe research at Georgia Tech designed to transform text-based requirements into SysML (Systems Modeling Language) model-based requirements representations, and vice versa. Active research in Systems Engineering is broad but has a common objective – improving the methods and outcomes for designing complex systems-of-systems. Advances in innovative technologies, algorithms, and software solutions will continue to benefit the practice of Systems Engineering.

The use of SysML for MBSE applications provides a common language and framework for development. As system complexities increase, so do the SysML models. Peterson (2015) describes a method using elements of Graph Theory coupled with MBSE to create visualizations that can aid the Systems Engineering teams in terms of comprehensiveness and implementation speed. The visualizations can extend beyond MBSE models to Systems Engineering documentation and specifications. Document similarity analysis and visualization methods can be used to aid reviews as well as the evaluation of Systems Engineering artifacts. Hussein (2016) defines a method using n-gram phrases, pair-wise matching, and latent semantic analysis to find and expose relationships among documents under consideration.

The Systems Engineering community can benefit from technological advancements in MBSE, AI, NLP, and data visualization. Multidisciplinary implementation of these elements can collectively benefit the community as modern systems continue to increase in terms of complexity, comprehensiveness, and digital persistence.

Natural Language Processing for MBSE

The use of NLP to aid MBSE is not a new idea. Prior research has shown that the main focal point of experimentation with NLP in the MBSE community has been centered around requirement structure and automatically converting requirements into SysML diagrams. Many applications and researchers are using NLP to evaluate requirements against common standards such as desired verb usage. However, our solution focuses on evaluating the relationships between various system components or aspects to support traceability activities.

Our NLP solution finds correlations between different aspects of a system to aid system traceability activities. The more complex the system, the more room for errors or missed traces. The duration of traceability activities is also directly affected by the complexity of the system being evaluated. Typical traceability activities can take hours, weeks, or even months to complete. The inclusion of NLP technologies is meant to reduce this time by acting as a first reviewer and taking a first attempt at tracing the different aspects. NLP technology is not meant to replace human analysts but enable them to complete tasks more thoroughly and in a timely manner.

This section outlines the experiment our team performed to demonstrate that an NLP algorithm can read two system documents and evaluating the relationships between them. For our experiment, we developed an NLP algorithm that can trace relationships between two requirements documents. Our experiment also enabled the MBSE professionals to save and export any desired trace relationships into Cameo, a system modeling tool from Dassault Systems. The experiment details and the results are outlined thoroughly in the following sections.

DEFINING THE EXPERIMENT

One of our first activities was to discuss user stories with MBSE professionals. Our team used these discussions to help guide and mold our experiment. Based on these discussions, our team was able to define a critical goal to accomplish: our experiment would utilize NLP to evaluate relationships between two different Systems Engineering documents. Based on further discussions with the MBSE professionals, we learned that these types of tracing activities are time consuming and occur regularly at different phases throughout a system's lifecycle. The frequency of these events makes our experiment more valuable as this technique can be used many different times throughout the system's lifecycle. With our experiment defined, it was time to create our algorithm.

Creating the Algorithm

For our purposes, NLP was applied to accomplish the mapping of relationships between disparate elements within and between documents. For this effort, we created and evaluated three different NLP algorithms.

The first NLP algorithm relied on a method known as Topic Modeling, specifically Latent Dirichlet Allocation. This approach attempts to build up a specified number of "topics" based on word clustering. Word clustering then builds lists of related, or clustered, words that make up topics. The idea was to then use these automatically created topics to classify the elements in the documents. However, we found that it was impossible to have substantial control over the topics that were generated. Many of the topics created too much noise which made the outputs difficult to interpret.

The second NLP algorithm then focused on an entirely different method of mapping. Rather than attempting to build topics and use those for classification, we decided to simply attempt to find the "similarity" between the elements directly. For this we combined a method known as Latent Semantic Indexing with a similarity matrix and cosine similarity to provide the desired outcome. Latent Semantic Indexing analyzes documents to extract statistical relationships between words that appear together. This means that the more the algorithm observes two words together, the more related those two words are. It also means that two different words, both appearing with a third word, will create a relationship or "bridge" between the two. All of these relationships are combined into a similarity score and presented to the user.

However, neither of these methods are new, nor are they cutting edge. Latent Semantic Indexing has been around since the late 1980s and has been applied heavily in products that are used every day, such as internet search engines. What is new, and the basis of a third NLP algorithm, is known as the transformer method. Transformers are a type of machine learning model that can, in general, be used for generative purposes. Specifically, they generate content. Transformers are behind much of the current buzz in the machine learning community; Chat-GPT is an example. For our experiment, we chose to fine-tune an existing open-source model known as RoBERTa. This model allowed for Sequence Classification, meaning we could create classes for each sequence that should be mapped (Liu et al., 2019). After the classes were created and fine-tuned, the process is as simple as providing the model with a piece of text to classify. A probability distribution is then returned, and the highest probability is used as the "truth."

Defining the Process and the Output

Based on the user stories our team gathered from the MBSE community, we defined the process shown in Figure 2. This process starts with the identification of the documents to be evaluated by the NLP algorithm and ends with the selected relationships being imported into Cameo.

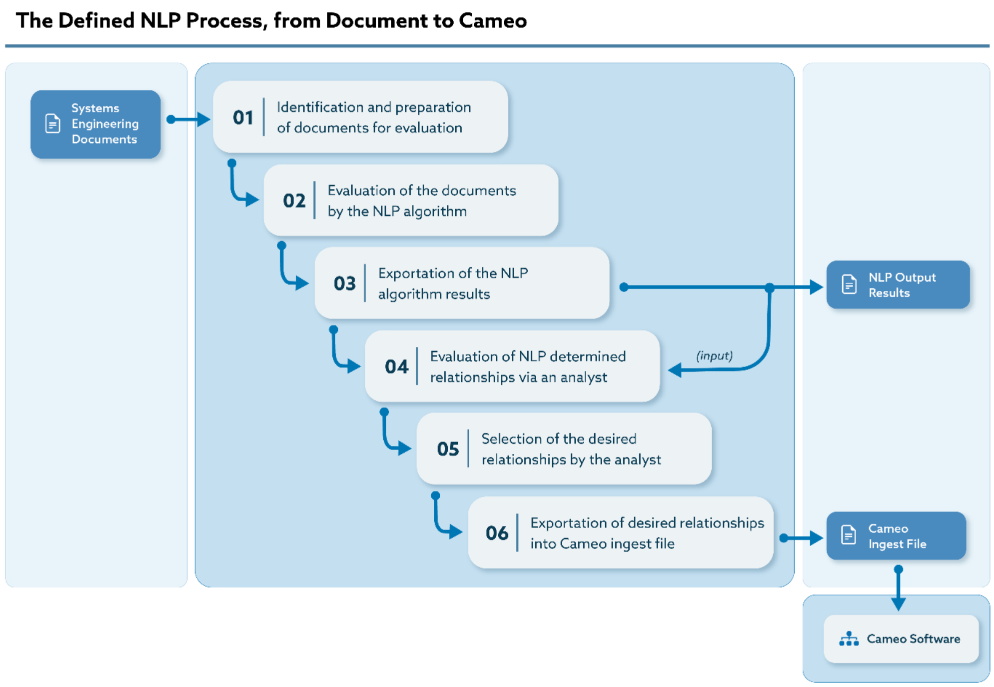

Figure 2. The Defined Process

After the process was determined, it was time to define the algorithm's output. To create both an effective and useful output, user needs were evaluated. Design decisions for the output were made based on input from the development team, MBSE professionals, and other key stakeholders in the experiment. Emphasis was placed on increasing usability of the output to promote analyst adoption.

At the conclusion of these user needs discussions, we decided to use Microsoft Excel as our output format. We chose this format based on its easy-to-read, table-like format that would make the results of the algorithm a lot easier to evaluate. This resultant Microsoft Excel file contains each of the input elements scored against the output elements. These relationships are given additional meaning through color coding, NLP degree of confidence, and a normalized score based on the probability score provided by the NLP algorithm.

Our Experiment and Cameo

Based on our discussions with MBSE professionals, there was a need to have findings imported easily into Cameo. Each of the users we interviewed regarding user needs utilized Cameo to perform their job responsibilities. Cameo allows users to create models that contain diagrams and relationships between various system aspects.

Based on discussions with users, creating these relationships manually in Cameo can be daunting and time consuming. Therefore, we added functionality to the NLP system to output a Microsoft Excel document that allows the user to click and select which relationships were to be exported into the Cameo ingestible file. This Cameo ingestible file could then be easily imported into any Cameo model. By being able to export the NLP algorithm's relationships into an ingestible Cameo format, the users can save time and effort. Our team tracked these metrics during our experiment.

Running the Experiment

After the process was defined, we needed to identify and prepare the documents for evaluation by the NLP algorithm. For this study, we selected two requirements documents to be evaluated. These requirement documents were created for a commercial, software-based project designed to view medical imagery in a 3D virtual environment. Our team chose these specific documents because they contain no customer or Government information, but they are representative of the scale and complexity of requirements documents for many of our MBSE projects.

The first document contained the system requirements for the medical application, while the second document contained the software requirements for the medical application. Based on the policies and procedures for the medical application and the requirement documents, the software requirements were to be derived from the system level requirements. This relationship of the requirements hierarchy is shown in Figure 3. These two documents have word and context relationships that should be revealed by the NLP algorithm.

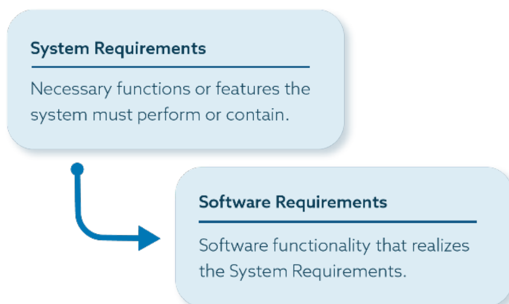

Figure 3. Requirements Derivation in the Documents

After the documents were chosen, care had to be taken to get the documents into a form that could be evaluated by the NLP algorithm. For this effort, a parsing algorithm was created to parse formatted Microsoft Word documents and subsequently ingest the contents of the Microsoft Word document into the NLP algorithm. In addition to being able to ingest formatted Microsoft Word documents, the NLP algorithm is also able to ingest Microsoft Excel documents.

After the documents were prepared for ingestion, they were then evaluated by the NLP algorithm and the output Microsoft Excel file was created that contained the results of the algorithm's evaluation. These results are discussed in the following section.

Results

We selected two datasets that outlined the requirements for the chosen medical software system. The two documents outlined both the system-level and software-level requirements for the medical system. For this project, the software-level requirements were derived from the system-level requirements. Therefore, there must be a relationship between any given software-level requirement and at least one system-level requirement. The algorithm evaluated the relationships between 70 system-level requirements and 206 software-level requirements. The NLP algorithm evaluated every system-level requirement against every software-level requirement. In this case, the NLP algorithm evaluated 14,420 unique relationships between these two sets of requirements. The NLP computational time is less than one minute. This is drastically less than the time it takes for a human to evaluate more than 14,000 requirements/requirement pairs. Based on discussions with our MBSE team, the average trace activity with similarly sized datasets takes several weeks to evaluate.

The NLP output is not a distinct answer. It provides a table of relationships and data that need to be reviewed by an MBSE practitioner. In our experiment, we used the software-level requirements as the base and then evaluated each of the software-level requirement's relationship to each of the system-level requirements in the second document. In doing so, the NLP algorithm provides a confidence score to each of the evaluated software-level requirements. This score indicates how confident the NLP algorithm is with the correlations it has identified for each software-level requirement. Each of the 206 software-level requirements are given a confidence score in the NLP algorithm's output.

In addition to the confidence value, the NLP algorithm also provides a score for each of the relationships evaluated. The scores range from 0 to 1, where 1 represents a strong correlation between the elements and 0 represents a weak correlation between the two elements. Table 1 shows how the scores relate to a strong, moderate, or weak correlation.

Table 1. Correlation Determination Based on Score

Correlation Determination	Score Range
Strong correlation	Score > 0.75
Moderate correlation	Score >= 0.25 AND Score < 0.75
Weak correlation	Score >= 0 AND Score < 0.25

During our experiment, we evaluated the confidence values of the NLP algorithm for each of the requirement correlations, the "correctness" of the relationships, and the time it took to evaluate the NLP algorithm's output results. To evaluate the "correctness" of the NLP algorithm, we used the project's requirements trace table as the ground truth and evaluated the NLP algorithm's traces against the traces contained in the table. The trace table contains all of the relationships between the system-level and software-level requirements for the project. These results are summarized in Table 2.

Table 2. Experiment Results

<table>
<tr><th colspan="3">Experiment Results</th></tr>
<tr><th>Category</th><th>Description</th><th>Value</th></tr>
<tr><td rowspan="2">Inputs</td><td>Number of evaluated software-level requirements</td><td>206</td></tr>
<tr><td>Number of evaluated system-level requirements</td><td>70</td></tr>
<tr><td rowspan="2">Confidence Values</td><td>% of high confidence requirement matches</td><td>46.12%</td></tr>
<tr><td>% of low confidence requirements matches</td><td>53.88%</td></tr>
<tr><td rowspan="3">Relationship Values</td><td>Number of unique relationships evaluated</td><td>14,420</td></tr>
<tr><td>Number of verified "correct" relationships</td><td>108</td></tr>
<tr><td>Number of relationships (from project requirements trace table)</td><td>378</td></tr>
<tr><td>Trace Table Coverage</td><td>% coverage of the project requirements trace table</td><td>28.57%</td></tr>
<tr><td>Time</td><td>Time taken to evaluate the NLP output</td><td>16 hours</td></tr>
</table>

Upon first evaluation of the raw results obtained in the table above, it can be inferred that the NLP algorithm performed poorly. The percentage of high confidence evaluations was less than 50%. The raw results also showed that only 108 of the strong correlations were verified as correct based on the project trace table. In fact, the NLP algorithm was only able to correctly identify 28.57% of the relationships in the project trace table as a strong correlation.

However, on further evaluation of the NLP algorithm's results, the software-level requirements the NLP algorithm had low confidence in were focused on specific user interface design. The system-level requirements that mapped to these user interface centered software-level requirements were vague and ambiguous. Therefore, it can be inferred that the NLP had difficulty matching these relationships due to ambiguity. In addition, the team identified that when evaluating both the identified strong correlations and the moderate correlations, the NLP algorithm covered 53.97% of the relationships identified in the project trace table. Table 3 displays the differences in results when evaluating the NLP algorithm's identified strong correlations only versus the results when evaluating both the identified strong and moderate correlations.

Table 3. Experiment Results Comparison

Experiment Results Comparison			
Category	**Description**	**Strong Correlations Only**	**Strong and Moderate Correlations**
Relationship Values	Number of verified "correct" relationships	108	204
	Number of relationships (from project requirements trace table)	378	378
Trace Table Coverage	% coverage of the project requirements trace table	28.57%	53.97%

In addition, the MBSE analyst found that of the 174 missed correlations, 131 of the missed correlations were associated with five system-level requirements. In other words, these five requirements accounted for 76.16% of the correlations missed by the NLP algorithm. Upon further evaluation of these five requirements, the team found that these requirements were ambiguous, catch-all requirements. The biggest offender required that the application be "interactable." This requirement accounted for 23.83% of the missed correlations. One reason for the missed correlations was the fact that none of the software-level requirements used the same wording ("interactable"). The software-level requirements described the user interface interactions in more detail. Therefore, the NLP algorithm was not able to make these correlations as it did not understand the correlation between the word "interactable," and the detailed interactions described in the software-level requirements.

Figure 4 shows the coverage of the project requirements trace table. The first pie chart depicts the breakdown of strong, moderate, and weak correlations identified by the NLP algorithm. The second pie chart shows the percentage of the NLP algorithm's weak correlations which are associated with the five ambiguous requirements. If these requirements were modified and the wording was improved, the NLP algorithm would likely have picked up on these correlations between the software-level requirements and the system-level requirements.

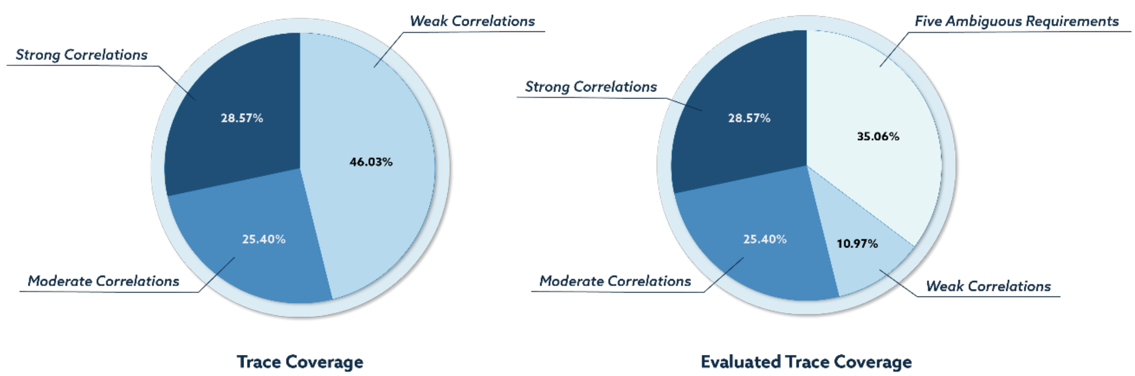

Figure 4. Project Requirements Trace Table Coverage Comparisons

By evaluating the missed correlations of the NLP algorithm, MBSE analysts can extrapolate areas of improvement to wording or even how they can improve requirement readability and therefore comprehension by a novice MBSE analyst. NLP algorithms draw many parallels to a novice MBSE analyst. Similar to a novice MBSE analyst, the NLP algorithm has no prior knowledge of the subject matter being discussed in the documents. The NLP algorithm is evaluating the relationships based on similar wording and concepts. However, for some of the missed correlations, the NLP algorithm could not easily understand words that can be used interchangeably because the training data was not sufficient. In other words, the NLP algorithm could not make the correlation between two words if it has only seen one of the words a few times. For example, the NLP algorithm was not able to make a correlation between a requirement that used the word "program" with a requirement that used the word "application." The reason was that all of the previous requirements consistently used the word "application." While a seasoned MBSE professional who knew the subject matter would be able to make this correlation between the words "application" and "program," the NLP algorithm was not able to make the correlation due its limited contextual understanding of the word "program." Similarly, a novice MBSE professional might also need further guidance to feel confident in making this type of correlation. This example shows that word consistency is important in improving readability and understanding when using NLP algorithms. In addition, being able to evaluate the content with an NLP algorithm to identify improvements in document wording can also promote greater understanding of the subject matter being evaluated. Our experiment focused on using NLP to find traceability between requirements in the two documents, but this finding highlights the potential value of using NLP to improve the Requirements Engineering process overall.

While not all of the traces in the project requirements trace table were found by the NLP algorithm, over half of the traces were identified. This trace and evaluation activity was performed in about two working days. Traditional manual trace activities with the same number of requirements can take weeks to perform. However, this experiment shows the NLP algorithm will not fully replace a human analyst as the algorithm was not able to fully replicate the project requirement trace table. Therefore, while the NLP algorithm proved to save time when determining 50% of the requirement coverages, the algorithm still needs to be augmented with human intuition. By allowing the NLP algorithm to be the first evaluator of the requirements, it can create the initial trace table in a fraction of the time it would take for a human evaluator to look through the same number of requirements, thereby increasing workflow for the Systems Engineering team.

CONCLUSIONS AND FUTURE WORK

In this paper, the authors outlined an experiment that traced relationships between two different but correlated sets of requirement documents. Our results indicated the effectiveness in applying NLP to the Systems Engineering domain. Upon evaluation of the results, the team determined areas of improvement to the documentation used for the experiment that could increase requirement readability and MBSE understanding of the requirements. The team also found that the use of an NLP algorithm could be used to increase workflow and decrease overall evaluation time.

There is more work to be done to enhance the capability of NLP for the DoD domain to aid in the validation of complex system architectures. While this study focused on evaluating the requirements of a 3D software system in the medical domain, we can extend the functionality of NLP to other domains. Every system is developed against a set of requirements that it is mandated to meet. These requirements can either be informal or formal. This common constraint enables NLP to enhance effectiveness and save evaluation time in any domain that applies Systems Engineering methodology. Regarding future work, our team will continue to perform experiments with requirement sets from several different systems across numerous domains.

In addition, the methodology described in this paper can also be applied to other Systems Engineering concepts. In future iterations of this research, our team plans to utilize the NLP capability in the MBSE space to extend to interfaces, functions, data, etc. As we expand our experiments, we will continue to leverage associated research to aid and validate our efforts.

REFERENCES

Bajceta, A., Leon, M., Afzal, W., Lindberg, P., & Bohlin, M. (2022). Using NLP Tools to Detect Ambiguities in System Requirements - A Comparison Study. *CEUR Workshop Proceedings, 3122*. **https://ceur-ws.org/Vol-3122/NLP4RE-paper-3.pdf**

Ballard, M., Peak, R., Cimtalay, S., & Mavris, D. (2020). Bidirectional Text-to-Model Element Requirement Transformation. *2020 IEEE Aerospace Conference*, 1-14. doi: 10.1109/AERO47225.2020.9172306.

Fuentes, J. M. (2019). Applying AI and NLP to enhance the definition of requirements. *Artificial Intelligence for Systems Engineering (AI4SE), 1*.

Hussein, A. S. (2016). Visualizing document similarity using n-grams and latent semantic analysis. *2016 SAI Computing Conference (SAI),* 269-279. doi: 10.1109/SAI.2016.7555994.

International Council on Systems Engineering (INCOSE). (2022). Systems engineering vision 2035, engineering solutions for a better world. INCOSE. **https://www.incose.org/docs/default-source/se-vision/incose-se-vision-2035-executive-summary.pdf?sfvrsn=ff2063c7_8**

Liu, Y., Ott, M., Goyal, N., Jingfei, D., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. **https://doi.org/10.48550/arXiv.1907.11692**

Madni, A. M. (2021). MBSE Testbed for Rapid, Cost-Effective Prototyping and Evaluation of System Modeling Approaches. *Applied Sciences, 11*(5)*,* 2321. https://doi.org/10.3390/app11052321

Peterson, T. (2015). Understanding Systems Through Graph Theory and Dynamic Visualization. *2015 NDIA Ground Vehicle Systems Engineering and Technology Symposium*. NDIA. **http://gvsets.ndia-mich.org/documents/SE/2015/Understanding%20Systems%20through%20Graph%20Theory%20and%20Dynamic%20Visualization.pdf**

Zhong, S., Scarinci, A., & Cicirello, A. (2023). Natural Language Processing for Systems Engineering: Automatic generation of Systems Modelling Language diagrams. *Knowledge-Based Systems, 259*. https://doi.org/10.1016/j.knosys.2022.110071