Natural Language Processing to Extract Contextual Structure from Requirements

Maximilian Vierlboeck
School of Systems & Enterprises
Stevens Institute of Technology
Hoboken, NJ
0000-0002-9518-1216

Daniel Dunbar
School of Systems & Enterprises
Stevens Institute of Technology
Hoboken, NJ
0000-0001-9647-8073

Roshanak Nilchiani, Ph.D.
School of Systems & Enterprises
Stevens Institute of Technology
Hoboken, NJ
0000-0002-1488-9934

***Abstract*—The automatic extraction of structure from text can be difficult for machines. Yet, the elicitation of this information can provide many benefits and opportunities for various applications. Such benefits have been identified amongst others for the area of Requirements Engineering. By assessing the Natural Language Processing for Requirement Engineering status quo and literature, a necessity for an automatic and universal approach to elicit structure from requirement and specification documents was identified. This paper outlines the first steps and results towards a modularized approach that splits the core algorithm from the text corpus as an input and underlying rule/knowledge base. This separation of functions allows for individual modification of the included parts and eases or potentially removes restrictions as well as limitations, such as input rules or the necessity for human supervision. Furthermore, contextual information and links via ontology inference can be considered that are not explicit on a textual level. initial results of the approach show the successful extraction of structural information from requirement text, which was validated by comparing the results to human interpretations for small and public sample sets. In addition, the contextual consideration and inference via ontologies is described conceptually. At the current stage, limitations still exist regarding scalability and handling of text ambiguities, but solutions for these caveats have been developed and are being tested. Overall, the approach and results presented will be integrated and are part of a novel requirement complexity assessment framework.**

Keywords—requirements engineering, natural language processing, complexity, structure, ontology, contextual information

I. Introduction

The extraction of structure and abstract information from a body of text can seem like a trivial task and drawing models, graphs, and networks as representations of information and thoughts can be an efficient way to communicate and bring certain insights to light. As straight forward as this task might seem though, automating or having it done even just in part by a machine can be difficult due various factors, such as subjectivity [1], ambiguity [2], and domain-specific circumstances. These factors can make universal application of tools difficult or require limitations to ensure function.

A popular approach to extract information and or structure from text is Natural Language processing (NLP). Despite sometimes controversial definitions of NLP [3], it is in its core the attempt to process natural language with computer tools that are supposed to allow a human-like linguistic analysis and manipulation of text/speech [3-5]. As such, extracting structure from text is one possibility of NLP. Yet, the research directions of NLP are manifold and thus, the aforementioned extraction of structure is a problem that can be approached in different ways. This variety lead to the development of numerous tools over time that could be used to extract certain types of structure from text or speech. As a result, the existence of solutions and application possibilities is not the problem as a plethora of tools, software, and ideas can be found, as show in Section II [6]. It is applicability and usefulness that are not always given, due to various limitations, for example. Such limitations can make not only the research of existing tools difficult since most have to be carefully assessed for their criteria, but also complicate the continuation of research due to the crowdedness of the space.

The extraction of information is especially critical when it comes to understanding contextual information that might not be explicitly part of the text. One application case of such an approach is Requirements Engineering in which human created natural language requirements and specifications are elicited and managed. To understand the underlying connections within requirements, relying on the text and its content alone is not expedient nor purposeful. As such, understanding the exact structure and connections behind the text layer requires a method such as NLP, and this application is what the presented research addresses.

By conducting an integrative literature [6], a research gap was identified, which shows that existing Natural Language Processing for Requirements Engineering (NLP4RE) tools have various limitations that make them either unusable for certain scenarios, or, due to a lack of open-source availability, only conceptually useful. Therefore, a new approach is being developed that allows for the mitigation of the limitations (see Section II for additional details).

To outline the work and process as well as results of the research, this paper has been divided into six sections. This first section introduces the field and situation, which is further expanded in the second section by an overview over the scientific field and results of the integrative review. Section III outlines the concept for the novel approach, which is demonstrated in Section IV and discussed in section V, including up to date results. The sixth and last section summarizes and concludes the paper before giving an outlook regrading future work and possibilities.

II. State of the Art and Literature

Since the topic at hand addresses NLP and the field of NLP4RE specifically, a brief history for the former and state of the art for the latter will be outlined hereinafter.

Natural Language Processing History and Progress

Looking at NLP from a general perspective, three domains emerge: *Linguistics*, *Computer Science*, and *Psychology* [3]. The first field, Linguistics, is concerned with the structural and formal aspects of language; the second one, Computer Science, focuses on the processing and structuring of data; and the last one, Psychology, contributes the insights into cognitive processes and psychological models of language. As a result, two directions exist in NLP: language processing and language generation. The language processing on one hand analyzes text/speech in order to create a representation, whereas language generation addresses the opposite: creating text from representation. The topic at hand is related to the former.

The historic origins of NLP date back to the 1940s, where Machine Translation (MT), now considered a precursor to modern NLP, was developed and explored [3, 4]. First MT descriptions go back to Weaver's article about translation from the year 1949 (later published as a book section in 1955) [7]. In the article, Weaver's thoughts on the possibilities and potential obstacles regarding the translation of languages by machines are outlined. MT began based on stochastic and statistical approaches that attempted to tackle issues such as different translations of words, meanings, and ambiguities.

Following the efforts from the 1940s and early 1950s, Chomsky published the idea of generative grammar in 1957 [8] as part of his "Syntactic Structures." The concept describes grammar as a certain set of rules that result in the constellations and combinations of words forming sentences in a given language. Chomsky breaks from popular theories of the time (e.g. Shannon's communication theory [9]) by saying that the structure of language cannot be addressed with pure statistical or empirical methods [10, 11]. In addition, Chomsky continued to work on aspects related to generative grammar all the way into the 1960s, [12] and his work ended up defining what is now considered the rationalist approach in NLP that was prominent until the mid 1980s [10, 13]. Furthermore, the concept is part of what is considered universal grammar that evolved over time with humans [13]. Figure 1 shows an example of the structure of a sentence according to Chomsky's approach and constellation: a sentence is divided into a noun phrase (NP) and an additional verb phrase (VP). The latter also includes the object as a noun phrase with the respective determiner.

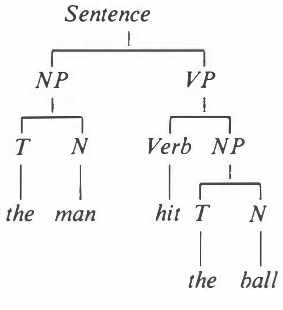


Figure 1 - Sentence construct according to generative grammar [8]

Throughout the 1960s, the movement based on Chomsky's approach of symbolic interpretation and the stochastic/ statistical one based on Shannon's methods [9] coexisted and advanced. Noteworthy results of this period are the first parsing systems by Harris [14] as part of the symbolic paradigm. The first mention of Artificial Intelligence (AI) in an NLP context occurred in the stochastic vein of NLP research [13] and Bledsoe and Browning [15] developed the first optical character recognition approach. Also in the 1960s, Woods published procedural semantics for a question-answering machine [16]. Albeit still based on programmed subroutines, Woods' publications show elements that can be associated with Natural Language Understanding (NLU) as the answer of a question requires the extraction of semantic meaning from the question. The application was limited, but question-answer machines are still used today in voice assistants.

During the 1970s and 1980s, the field of NLP grew broader and topics such as NLU emerged, which introduced elements such as text/speech recognition and synthesis [17, 18]. NLU was first demonstrated by Winograd [19]. In the publication "Understanding natural language" the authors show a program that is able to identify and select different shapes and colors in a simulated environment based on given text commands. This work bears strong ties with Woods' work mentioned above [16], and both drove the field of logic-based NLU. Additional noteworthy contributions to this trend include Schank and his colleague's work on language understanding programs [20-22].

In the second half of the 1980s and early 1990s, statistical approaches re-emerged [23] as the primary focus of NLP/NLU, moving away from the symbolic ideas shaped by Chomsky [10, 12]. This popularity of stochastic methods in speech/ language processing was significantly driven by IBM's Thomas J. Watson Research Center [13]. The re-emergence came with novel speech-recognition models that sought to bring NLU and speech analysis closer together [24]. Eventually, before the beginning of the twenty-first century, the described changes and refocused popularity had made probabilistic models the predominant force in NLP, and the rapid increase in computing power, as well as the expansion of the internet, created a need for language-based information processing [13]. These circumstances lead to a more unified but changed field of NLP/NLU and eventually gave way to the rise of Machine Learning in the twenty-first century.

In the last 20 years, the interest in NLP has further increased in conjunction with the adoption of Machine Learning [18]. The pace that the subject had picked up by the end of the 1990s was unprecedented, [13] especially since the developments before were described as incremental [25]. As a result, numerous datasets were published in a few years [26-28]. These sets were collections that contained text structures with underlying semantic information about syntactics. With the help of such datasets, further advances in parsing, tagging, reference resolution [29], and information extraction were enabled [13]. In addition to the published sets, ML applications incorporated models such as the Bayesian Analysis [30] and maximum entropy to train systems to process text in accordance with semantic, morphological, and or syntactic parameters [29]. Notable results were significant improvements for some of the aforementioned, such as disambiguation, answering of questions by a machine and summarization [29]. Still today, computer linguistics in total is described as an active field in AI research [31, 32].

In summary, NLP has gone though various changes over time. It began with machine translation and stochastic approaches, then transitioned to semantic and symbolic methods. A broader expansion accompanying the emergence of concepts of NLU and speech recognition enabled regained popularity of stochastic approaches before rapid changes in computer hardware and expansion of the web supercharged the progress of NLP, NLU, speech recognition, and machine translation, that are now being propelled by ML and AI. Also, potential future developments have been explored and considered as shown in Figure 2 based on the predictions by Cambria and White [33] who predict less reliance on word-based techniques to utilize semantics more effectively:

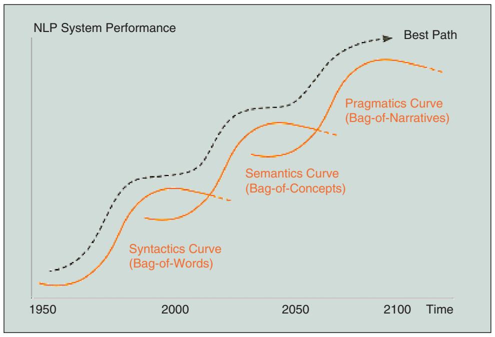


Figure 2 - Considered Evolution of NLP over Time [33]

Natural Language Processing for Requirement Engineering

Based on the situation described in the introduction, the approach presented pertains to the combined field of NLP together with Requirements Engineering (RE), which is called NLP for RE (NLP4RE). For this area, various approaches exist with some going back to the time that can be considered the mainstream beginning of RE in the late 1990s [34]. This long history and have made the field very diverse. As a result of this diversity, looking for different approaches can be difficult as not all of them achieve popularity due to their niche existence and/or special purpose applications. Fortunately, studies have been conducted that target this issue. The most comprehensive one to date was published by Zhao et al. [34] in 2021. This study assessed the space of NLP4RE regarding tools and solutions. The results were 404 relevant studies that the authors classified, from which they extracted 130 tools [34]. In addition, Zhao et al. emphasize that most of the tools have not made it out of laboratory settings, are focused on the analysis of requirements, and require specifications [34, 35], which is in line with the research at hand. Furthermore, the results and insights presented by Zhao et al. also have been confirmed by research of the authors, whose work and discovered results turned out to be overlapping sub-sets. [36, 37].

To add to the foundation above, additional publications show current trends and activities: the following publications were identified as applicable to the task at hand. First, Mengyuan et al. presented an approach that utilizes NLP to extract domain models for control systems [38]. Their approach is based on Rupp's template for requirements and allows for the extraction as well as visualization of models. Second, two tools addressing causality in requirements and the detection thereof were discovered. These tools, CiRA [39] & CATE [40], address causal relationships within requirements. These relationships are assessed as to which requirement causes or depends on others. Third, Sonbol, Rebdawi, and Ghneim published their approach called ReqVec that allows for the deduction of semantic relationships as well as classification of requirements [41]. This approach, based on Word2vec showed a high efficiency in tests. Fourth, Schlutter and Vogelsang published their approach to trace the connections between requirements, which they call Trace Link Recovery [42]. This approach utilizes an explicit content description of the requirements in the form of a semantic relations graph that allows for the tracing of connections within. Lastly, van Vliet et al. [43] present an approach for NLP crowdsourcing to solve shortcomings regarding a lack of accuracy and reliability of current approaches.

All in all, the previous paragraph indicates active and ongoing research in the field of NLP4RE. Furthermore, the different directions show that there are still various topics and ideas being pursued. This further supports the purpose of the presented work, as structure and additional organization are valuable. Also, such structure can contribute to currently identified challenges, as outlined by Kaddari et al. [44].

Integrative review of the NLP4RE field

With the field and literature above, the potential exists that tools addressing the research gap described in the introduction are available. To determine the applicability of existing approaches, they were assessed to evaluate if they are applicable and address the problem at hand. This assessment was conducted in the form of an integrative review by Vierlboeck, Nilchiani, and Lipizzi [6]. For the evaluation, criteria to determine the suitability of approaches were used to assess available options. This also enabled a better understanding regarding the diversity of the NLP4RE space.

The insights resulting from the integrative review were that no approach fulfilled all defined criteria and could thus be deemed applicable. The contenders that satisfied most criteria only targeted the extraction of structure in part and thus require further adaptation to be useful. Other approaches/tools that target the elicitation of structure turned out to not be accessible as far as their code base is concerned, which makes them only useful conceptually and thus require reconstruction. All in all, the analysis showed that no existing approach in the public space addresses the problem and situation set forth in the introduction and hence, the creation of a novel tool to tackle this task is auspicious.

Despite the confirmation of the purpose introduced, further insights were gained from literature. For one, the statistic results of the integrative review show a high reliance of the tools on supervision and input requirements, which has to be kept in mind in the work moving forward. Such a wide-spread reliance on supervision and input limitations can indicate the general difficulties with unrestricted, automatic, and universal approaches, which could mean that initial limitations or general restrictions might have to be considered. Second, the data and sources also show that concept proofs and validations are often only conducted in a theoretical manner, which supports the laboratory setting claim by Zhao et al. [33].

All in all, the outlined work shows the purpose of a universal approach to elicit structure from requirements and as such, this paper presents the first steps to develop a framework for that purpose.

III. Novel NLP4RE approach for structure elicitation

In order to develop a framework for the elicitation of structure via NLP, the design research methodology by Blessing & Chakrabarti [45] was used based on the insights and literature above. With this, a solution was developed that addresses limitations, such as input or supervision conditions.

Furthermore, since the literature showed that many solutions were not easily transferrable, a modularization of the approach was developed. This modularization divides the methodology into three pillars: the algorithm, the corpus/input, and the knowledge base. Herein, the corpus represents the requirement document as depicted below.

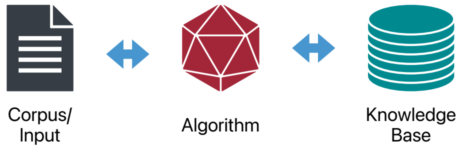


Figure 3 - Modularization of NLP4RE Approach

By splitting up the approach into three pillars, various advantages arise: for one, each part can be modified individually; for two, the algorithm/function becomes independent from the knowledge base as it is not intertwined with the content of the latter; for three, modifications to each of the three components are less likely to cause problems as long as the interfaces don't change. Furthermore, modularization enables one of the most important flexibilities as the separation of the knowledge base makes the latter an exchangeable part. As such, the adaptation of the entire approach to different circumstances or domains can be achieved.

To develop/choose the separate components, they were first addressed individually, starting with the algorithm. This was then brought together with the input to test it and finally, the knowledge base was brought in. This last pillar is expected to adapt over time due to its exchangeable nature.

The entire development was implemented in Python with the spaCy resources in the next section as well as the ontology software Protégé. Furthermore, the most recent results are included below.

Core algorithm and process

The core algorithm of the approach contains various NLP tools that are being used in tandem to achieve the extraction of structure. Fortunately, requirements as well as the documents they form have certain rules if they are phrased correctly. This can be due to the correct application of standards, such as the ISO/IEC/IEEE International Standard 29148 [46, 47]. These rules potentially increases the effectiveness of the extracted information but were not used as a guarantee for correctness, as discussed in more detail below.

Specifically, the tools and NLP parts or spaCy used are: tokenization, splitting, part-of-speech tagging, dependency parsing, lemmatization, chunking, and entity linking. By connecting these tools, two main objectives were achieved: first, the information within the requirement text was reduced to the structural parts, and second, the certainty of the identified information could be indicated. Therefore, the flow depicted in Figure 4 was followed.

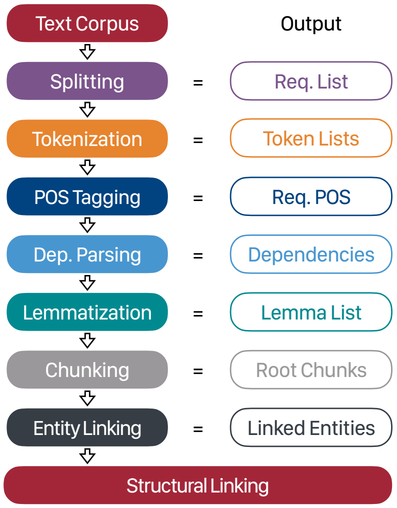


Figure 4 - Flowchart

In the first step, the text is split into its sentences by dividing it according to punctuation and identifiers. Next, the individual sentences are tokenized. Then, based on the tokens together with the respective sentence they are in, part-of speech tagging and dependency parsing is processed. These two steps in conjunction yield an important result: the role each token has in the sentence and how they depend on each other. With this information, the entities in the sentences as well as their connections are identified. For instance, in a noun-sentence, the nominal subject forms the acting entity related to (if existing) the object via the verb or root. In the simplest cases, these three pieces form a triplet of subject, predicate, and object, which is of importance for the inclusion of the contextual information.

Lastly, to ensure consistency and avoid potential errors due to verb tenses, for instance, lemmatization is used to reduce each identified piece to its core form, allowing the subsequent steps to not only check the actual content, but its roots as well. This last step lead to the output shown in Table 1, which is then used to derive structure and include contextual information.

#	Entity	Connection	Object	Check
1	laptop	have	display	✓
2	laptop	have	storage	✓
3	chassis	made	material	✓
4	display	have	resolution	✓
5	storage	have	redundancy	✓

Table 1 - Exemplary Output Table

As seen in Table 1, the output already has a structure to it as it contains the identification and setup of each requirement. Yet, there is no cross-connection of the individual requirements or entities therein due to the fact that they are being processed separately. These cross-connections and interrelations are being addressed in a subsequent step, which includes the consideration of context as explained below.

As Table 1 shows, the output of the NLP sequence are identified entities of each requirement statement. These entities are the base of the connections in said statement. Directly linked to these entities are the object. The connection between these identified parts is the linking verb or chunk, that also includes directional information as a result. Lastly, the check column indicates the absence or presence of ambiguities. In case of more than one identified element for each of the categories or columns, the check variable is used to indicate a possible mis-match or omission of information. Within the code, ambiguities for every category are stored individually and as a result, ambiguities can be tracked down if they occur, which simplifies addressing the issue (also see Figure 8). Lastly, it shall be noted that a certain term or chunk of words can be entity and object for different requirements. For instance, the object 'display' in Table 1 is the object of the laptop entity (line 1), but functions as the entity for the 'resolution' object in line 4. These connections and sub-connections are what eventually forms the network/structure, together with the context connections in the next sub-section.

Elicitation of structure with consideration of context

With the entities/objects extracted from the requirements, a structure identification and elicitation is already possible, but the contextual aspects of the requirements are not yet considered. Contextual connections are the implicit links that might not be explicitly stated in the requirement text. As such, these connections can stem from various sources. For instance, different words used or changed expressions can lead to missing links that otherwise should be included and are potentially crucial for the structure. In addition to the usage of different terms, crucial connections can exist between requirements that are inferred but not visible on the text layer. An example for these hidden links are requirements that relate to certain aspects of the system without explicitly mentioning said relation. This is in part due to the rigid structure of the statements, but also due to human context inference, which is not considered with the explicit text layer processing. Figure 5 shows an example for the difference between explicit and implicit connections.

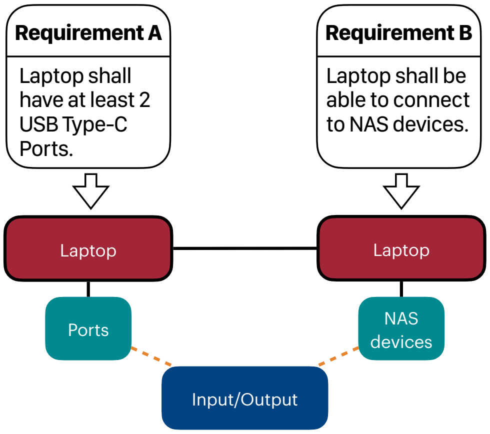

Requirement A
Laptop shall have at least 2 USB Type-C Ports.
Requirement B
Laptop shall be able to connect to NAS devices.
Laptop
Laptop
Ports
NAS devices
Input/Output

Figure 5 - Implicit Connection Illustration

As seen in Figure 5, the two example requirements allow for the correct elicitation of the entities. Also, since they share the same subject entity, a connection on the top level can be derived. Yet, the objects on both sides are different in text and meaning, which does not allow for any link on said level based on explicit information. Yet, the two requirements both pertain to the 'Input/Output' capabilities of the system, although they connect to it in a different way. As a result, the context, although shared, cannot be elicited on a textual level without inference. The addition of this elicitation is the second part of the presented approach. This inclusion of context is accomplished by adding information in the form of the knowledge base. As such, this addition will allow for the inference of the connections not determinable based on text.

By considering the context and implicit links between the elements of the structure, the network can be expanded to also show these additional links. An example for such an addition is shown in Figure 6: two entities, A and B have multiple object / child nodes, but are not connected through any direct/explicit links. Yet, through the inclusion of context and implicit information, the branches stemming from entity A & B become connected as they share context A & B. These connections can be crucial as they bring together previously separate networks and thus allow for a more comprehensive analysis that was impossible for each structure individually.

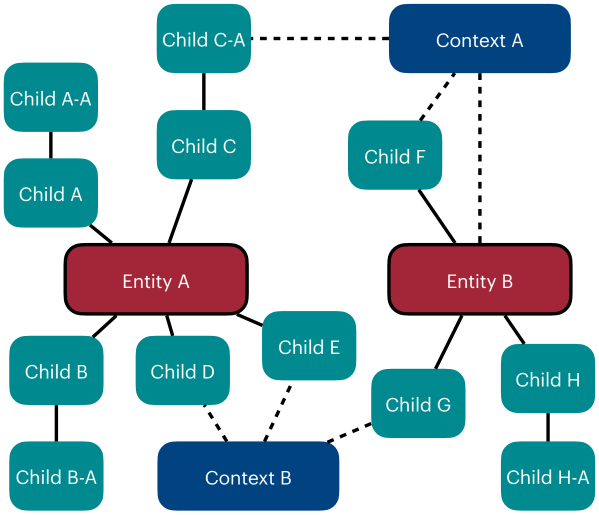


Figure 6 - Implicit & Explicit Connection Network Example

For the provision of the contextual information in form of the knowledge base, an ontology application was chosen. As such, alignment of requirement structure and entity identification with an ontology will be described hereinafter. In general, ontologies contain formal representation of knowledge and the relationships between entities beginning with a subclass taxonomy and expanding over additional relationships such as part_of, describes, and prescribes. Ontologies can be structured using the Web Ontology Language (OWL), which is based on Description Logics. The use of formal logic allows for automated inference of new knowledge based on existing entities and relationships within the ontology [48].

For example, a document lists the following requirements:

- The laptop shall have a solid state storage device.
- The laptop shall have a backup disk drive storage device.

Elsewhere in the specification document, another requirement is found, reading as follows:

- The system shall utilize commercial off-the-shelf (COTS) storage devices.

The last requirement has an inferred relationship with the first two. Since each refers to types of storage devices, the latter requirement puts a constraint on the initial requirements. An ontological representation for this example is presented in Figure 7.

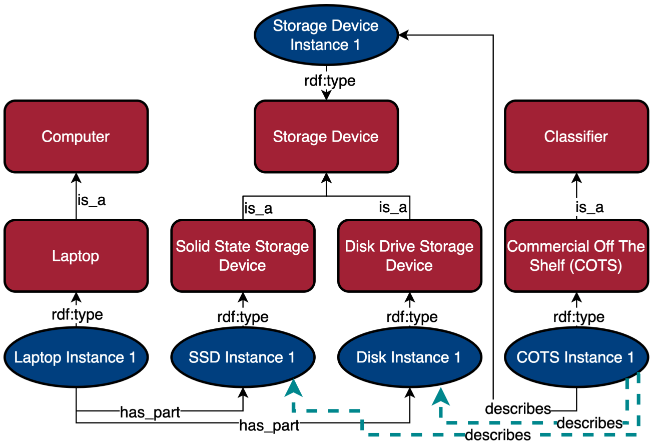


Figure 7 - Ontological Representation

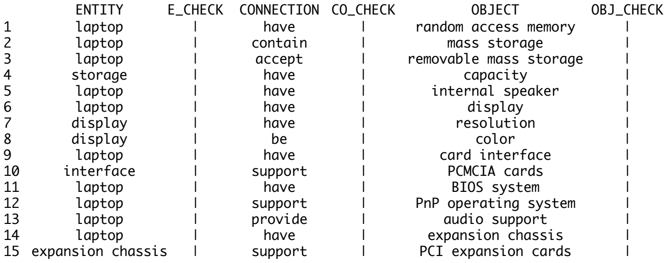
The rectangles are classes defined in the domain ontology and show a basic taxonomy. The instances that are mapped from the requirements documents are shown in ovals, and the relationships established by the NLP algorithm are shown. For instance, the 'has_part' is the relationship established by the requirements between 'Laptop Instance 1' and 'SSD Instance 1'. From these relationships, a DL reasoner and a rule written in the Semantic Web Rule Language (SWRL) can infer that 'COTS Instance 1' describes the two storage device instances that are part of the laptop (shown by the dashed line) and as such, provide contextual information that is implicit. This implicit connection is possible due to the existing information of the ontology in combination with the NLP results. This example of reasoning demonstrates the power a formal representation of the domain knowledge can bring when requirements are mapped to an ontological representation.

In conclusion, the combination of the NLP processing with the added inference and context consideration via the ontology allows for the contextual elicitation of structure from requirement text. To demonstrate the application of the developed approach, the next section will provide sample results to illustrate the capabilities and opportunities.

IV. Results and Validation

With the concept and methodology described in the previous section, tests were conducted with sample sets of requirements that were openly accessible. These samples were obtained from public sources such as NASA and are available online [for example, see 49]. With these examples, the developed approach could be tested on various sets of requirements, which also allowed for the inclusion of different levels of quality as far as the input is concerned.

By applying the NLP process to the input of the requirement lists in textual form, the methodology was applied as outlined in the previous section. The processing of the text input and corpus, depicted in Figure 4, produces the tabular output of the identified objects and entities. An example for this output based on the sample set [49] is depicted in Figure 8.



```
        ENTITY      E_CHECK  CONNECTION  CO_CHECK         OBJECT         OBJ_CHECK
1       laptop         |       have         |      random access memory      |
2       laptop         |      contain       |         mass storage           |
3       laptop         |      accept        |    removable mass storage      |
4       storage        |       have         |           capacity             |
5       laptop         |       have         |       internal speaker         |
6       laptop         |       have         |            display             |
7       display        |       have         |          resolution            |
8       display        |        be          |             color              |
9       laptop         |       have         |        card interface          |
10     interface       |      support       |         PCMCIA cards           |
11      laptop         |       have         |          BIOS system           |
12      laptop         |      support       |      PnP operating system      |
13      laptop         |      provide       |         audio support          |
14      laptop         |       have         |       expansion chassis        |
15 expansion chassis   |      support       |      PCI expansion cards       |
```

Figure 8 - Sample Output

As seen in Figure 8, the output of the NLP process matches the expected output of the overview in Table 1 (with three individual check columns) and allows for subsequent processing. To validate the results, the elicited information has been cross-checked with human application of the defined rules to validate the output through logical reasoning. These tests have shown that the output produced is as expected and besides the limitations discussed in the fifth section, the results are correct and present a solution for the gap outlined in the introduction as well as the integrative review.

As for the inference via the ontology, function tests in Protégé [50], have been successfully conducted and show the possibility of the implementation as described in the previous section. The combination with the NLP process is pending as of the time of this writing (February 2022), but is being worked on and will be validated as part of the scalability efforts.

The NLP process results have been successfully translated into visual structures by creating a network of the connections via the library 'NetworkX' [51]. This has been achieved by using the entity and object connections and creating a network from the table of the NLP output. Such a network then represents the connections within the text itself and can then be combined with the ontological connections, which are currently being implemented. Since such a visual structure is only one possible representation of the gained insights, others are currently being evaluated as specific information aspects are omitted by network-only representations, such as the directional aspects of verb connections also elicited from the text. This shows that the created approach and tools show a multitude of possible expansion and transfer opportunities in addition to their current merits. To clearly outline the presented contributions, Section V also includes current limitations.

V. Discussion and Limitations

Looking at the results presented in the previous section, the possibility provided by the algorithm to extract structure from requirement text can be considered valid and on a small scale and as proof of concept. As for the interpretation of contextual inference via the knowledge base in the form of an ontology, Section III has shown the possibility and opportunities, but due to the necessity to develop sufficient and domain specific ontologies, valid tests and results that programmatically connect an ontological database to the NLP algorithm have not been produced yet. Nevertheless, the implementation is possible and is actively being worked on with requirement sample sets and compatible ontological foundations to show the programmatic possibilities as well to comply with the research gap and solution plan. As such, the task at hand and research presented contributes to the gap and problems outlined in the introduction. Yet, some limitations remain, which shall be discussed hereinafter.

First, as shown in the results above, ambiguities can be a major issue for machines to deal with compared to humans. This can be in part due to the fact that requirements originate in most cases from humans and as such are subject to flaws and errors. As a result, a high dependency on input quality exists as far as the requirement text corpus/input is concerned. While this dependence does not completely inhibit the functionality of the framework, it has to be considered and addressed moving forward as otherwise the automatic/universal nature of the objective cannot be achieved. Further, parallel efforts by other researchers, practitioners, and educators to increase the quality of written requirements should continue to be pursued.

Second, as already alluded to above, the presented inclusion and interpretation of context is dependent on the knowledge base, in this case the ontology. If such a foundation does not exist or is small in size, the inferred connections will provide only little to no additions to the structure and as such, a dependency on a sound knowledge base exists. Furthermore, the connections to be considered are another topic that is being evaluated as domain specificity has to be kept in mind when it comes to the knowledge base, as outlined by Lipizzi et al. [1]. Similar considerations are part of the ontological research field and will be included moving forward.

Third and last, the small size of the utilized samples poses a limitation. Small sets of requirements, while representative and applicable for the purpose of this research, are self-contained to some extent and elements within them can be assumed to be associated with unique entities/elements. Yet, this assumption does not necessarily hold for large sets as the same term might be identified more than once while referring to different elements. As a result, larger sample sets will require additional distinguishing aspects and consideration. As a consequence, the scalability of the approach, while not limited from a conceptual perspective, will come with additional challenges that need to be addressed, such as document hierarchy and organization. The splitting of information based on these factors is currently being investigated together with larger sample sets.

Despite the limitations, the research presented can be seen as a conceptually valid proof that shows opportunities for the extensions that are being worked on.

VI. Summary and Conclusion

The paper at hand presented the current gap in the scientific field of Requirements Engineering that concerns the extraction of structure from textual requirements also considering context and implicit connections. Such connections can be crucial to the structure of a system, but due to their lack of direct expression in textual form are difficult to elicit on a language level without inference. These circumstances have further been stressed by the integrative review results of the NLP4RE space [6], which showed that most tools available in this space come with limitations that make them either inaccessible or not fully compatible with the task at hand. Furthermore, most available tools come with additional input or supervision limitations. As a result, the paper presents a novel approach to utilize Natural Language Processing and ontologies to extract contextual structure from requirements by splitting the algorithm from the text corpus/input and a knowledge base that provides the necessary input for the context. To realize the implementation, several NLP tools in conjunction were employed with the addition of an ontology inference process that allows for the inclusion of contextual and implicit information.

By applying the created approach, results were produced that showed the possibility of the successful elicitation of contextual structure from requirement text based on sample sets that were openly accessible [49]. The current results have proven to be accurate compared to information/structure elicited through human logical reasoning. This shows the potential that automatic extraction of structure is possible with the created approach. Due to the early stages of the development, the approach does have some limitations. Mainly, the results and validity of the process are still dependent on the input quality, for instance requirement accuracy and absence of ambiguities. In addition, the inclusion of context has been shown conceptually with functional proof, but will require additional implementation to allow for an application transferrable to other problems. Nevertheless, the mentioned limitations are planned to be addressed with future work by further expanding the capabilities of the algorithm and knowledge base/ontology. Also, more work is planned and has already begun regarding the input quality dependance. All in all, the presented work shows the promising possibility and concept that can fill the research gap outlined without severe limitations, such as input restrictions or full supervision. Furthermore, the use of open-source code base for the algorithm can be shared openly upon request.

Acknowledgment

The research and results presented in this publication were supported by the Systems Engineering Research Center (SERC). SERC is a university affiliated research center managed by Stevens Institute of Technology. The authors would like to thank SERC and Professor Dinesh Verma for their support in making the presentation and publication of this work possible.

References

[1] C. Lipizzi, D. Borrelli, and F. Capela, "The "Room Theory": a computational model to account subjectivity into Natural Language Processing," arXiv.org, 2020, arXiv:2005.06059.

[2] D. M. Berry, E. Kamsties, M. M. Krieger, W. L. S. Lee, and W. L. S. Tran, "From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity," 2003.

[3] E. D. Liddy, "Natural Language Processing," in *Encyclopedia of Library and Information Science*, 2nd ed.: NY. Marcel Decker, Inc., 2001.

[4] G. G. Chowdhury, "Natural language processing," *Annual Review of Information Science and Technology,* vol. 37, no. 1, pp. 51-89, 2003, doi: https://doi.org/10.1002/aris.1440370103.

[5] T. Beysolow, "What Is Natural Language Processing?," in Applied Natural Language Processing with Python : Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing. Berkeley, CA: Apress, 2018, pp. 1-12.

[6] M. Vierlboeck, C. Lipizzi, and R. Nilchiani, "Natural Language in Requirements Engineering for Structure Inference - An Integrative Review," 2022, arXiv:2202.05065.

[7] W. Weaver, "Translation," in *Machine Translation of Languages*, W. N. Locke and A. D. Booth Eds. Cambridge, MA: MIT Press, 1955.

[8] N. Chomsky, *Syntactic Structures*. Mouton & Co., 1957.

[9] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal,* vol. 27, no. 3, pp. 379-423, 1948, doi: 10.1002/j.1538-7305.1948.tb01338.x.

[10] C. Manning and H. Schütze, Foundations of Statistical Natural Language Processing. Cambridge, MA: MIT Press, 1999.

[11] D. A. Dahl, "Natural Language Processing: Past, Present and Future," in *Mobile Speech and Advanced Natural Language Solutions*, A. Neustein and J. A. Markowitz Eds. New York, NY: Springer New York, 2013, pp. 49-73.

[12] N. Chomsky, *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press, 1965.

[13] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2008.

[14] Z. S. Harris, *String Analysis Of Sentence Structure*. The Hauge, Netherlands: Mouton Publishers, 1962.

[15] W. W. Bledsoe and I. Browning, "Pattern recognition and reading by machine," presented at the Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference, Boston, Massachusetts, 1959. [Online]. Available: https://doi.org/10.1145/1460299.1460326.

[16] W. A. Woods, "Procedural semantics for a question-answering machine," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, San Francisco, California, 1968: Association for Computing Machinery, pp. 457–471, doi: 10.1145/1476589.1476653.

[17] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[18] T. Beysolow II, "What Is Natural Language Processing?," in Applied Natural Language Processing with Python : Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing. Berkeley, CA: Apress, 2018, pp. 1-12.

[19] T. Winograd, "Understanding natural language," *Cognitive Psychology,* vol. 3, no. 1, pp. 1-191, 1972, doi: 10.1016/0010-0285(72)90002-3.

[20] R. C. Schank, "Conceptual dependency: A theory of natural language understanding," *Cognitive Psychology,* vol. 3, no. 4, pp. 552-631, 1972, doi: 10.1016/0010-0285(72)90022-9.

[21] R. C. Schank and R. P. Abelson, *Scripts, plans, goals and understanding: An inquiry into human knowledge structures* (Scripts, plans, goals and understanding: An inquiry into human knowledge structures.). Oxford, England: Lawrence Erlbaum, 1977.

[22] R. C. Schank and C. K. Riesbeck, *Inside Computer Understanding*. New York, NY: Psychology Press, 1981.

[23] K.-F. Lee and R. Reddy, Automatic Speech Recognition: The Development of the Sphinx Recognition System. New York, NY: Springer Science+Busienss Media, 1988.

[24] L. Hirshman, "Overview of the DARPA Speech and Natural Language Workshop," presented at the Proceedings of the workshop on Speech and Natural Language, Philadelphia, PA, 1989.

[25] Futures Group Glastonbury CT, "State of the Art of Natural Language Processing," Defense Technical Information Center, ADA188112, 1987.

[26] M. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank," University of Pennsylvania Department of Computer and Information Science, MS-CIS-93-87, 1993.

[27] J. Pustejovsky et al., "The TIMEBANK Corpus," in Proceedings of the Corpus Linguistics Conference, 2003, pp. 647–656.

[28] M. Palmer, D. Gildea, and P. Kingsbury, "The Proposition Bank: An Annotated Corpus of Semantic Roles," *Computational Linguistics,* vol. 31, no. 1, pp. 71-106, 2005, doi: 10.1162/0891201053630264.

[29] R. Kibble, *Introduction to natural language processing*. London, United Kingdom: University of London, 2013.

[30] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*, 3rd ed. Boca Raton, FL: CRC Press, 2013.

[31] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing [Review Article]," *IEEE Computational Intelligence Magazine,* vol. 13, no. 3, pp. 55-75, 2018, doi: 10.1109/MCI.2018.2840738.

[32] E. Ghazizadeh and P. Zhu, "A Systematic Literature Review of Natural Language Processing: Current State, Challenges and Risks," in *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1*, Cham, K. Arai, S. Kapoor, and R. Bhatia, Eds., 2021// 2021: Springer International Publishing, pp. 634-647.

[33] E. Cambria and B. White, "Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]," *IEEE Computational Intelligence Magazine,* vol. 9, no. 2, pp. 48-57, 2014, doi: 10.1109/MCI.2014.2307227.

[34] L. Zhao *et al.*, "Natural Language Processing for Requirements Engineering: A Systematic Mapping Study," *ACM Comput. Surv.,* vol. 54, no. 3, p. Article 55, 2021, doi: 10.1145/3444689.

[35] A. Ferrari, L. Zhao, and W. Alhoshan, "NLP for Requirements Engineering: Tasks, Techniques, Tools, and Technologies," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 25-28 May 2021 2021, pp. 322-323, doi: 10.1109/ICSE-Companion52605.2021.00137.

[36] A. Alzayed and A. Al-Hunaiyyan, "A Bird's Eye View of Natural Language Processing and Requirements Engineering," *International Journal of Advanced Computer Science and Applications,* vol. 12, no. 5, pp. 81-90, 2021, doi: 10.14569/IJACSA.2021.0120512.

[37] H. Schrieber, M. Anders, B. Paech, and K. Schneider, "A Vision of Understanding the Users' View on Software," in *Joint Proceedings of REFSQ-2021 Workshops, OpenRE, Posters and Tools Track, and Doctoral Symposium*, Essen, Germany, F. B. Aydemir *et al.*, Eds., 2021.

[38] Y. Mengyuan *et al.*, "Automatic Generation Method of Airborne Display and Control System Requirement Domain Model Based on NLP," in *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*, 23-26 April 2021 2021, pp. 1042-1046, doi: 10.1109/ICCCS52626.2021.9449277.

[39] J. Fischbach *et al.*, "Automatic Detection of Causality in Requirement Artifacts: The CiRA Approach," Cham, 2021: Springer International Publishing, in Requirements Engineering: Foundation for Software Quality, pp. 19-36, doi: 10.1007/978-3-030-73128-1_2.

[40] N. Jadallah, J. Fischbach, J. Frattini, and A. Vogelsang, "CATE: CAusality Tree Extractor from Natural Language Requirements," arXiv.org, 2021, arXiv:2107.10023.

[41] R. Sonbol, G. Rebdawi, and N. Ghneim, "Towards a Semantic Representation for Functional Software Requirements," in *2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, 1-1 Sept. 2020 2020, pp. 1-8, doi: 10.1109/AIRE51212.2020.00007.

[42] A. Schlutter and A. Vogelsang, "Trace Link Recovery using Semantic Relation Graphs and Spreading Activation," in *2020 IEEE 28th International Requirements Engineering Conference (RE)*, 31 Aug.-4 Sept. 2020 2020, pp. 20-31, doi: 10.1109/RE48521.2020.00015.

[43] M. van Vliet, E. C. Groen, F. Dalpiaz, and S. Brinkkemper, "Identifying and Classifying User Requirements in Online Feedback via Crowdsourcing," Cham, 2020: Springer International Publishing, in Requirements Engineering: Foundation for Software Quality, pp. 143-159.

[44] Z. Kaddari, Y. Mellah, J. Berrich, M. G. Belkasmi, and T. Bouchentouf, "Natural Language Processing: Challenges and Future Directions," Cham, 2021: Springer International Publishing, in Artificial Intelligence and Industrial Applications, pp. 236-246.

[45] L. T. M. Blessing and A. Chakrabarti, *DRM, a Design Research Methodology*. London, United Kingdom: Springer-Verlag, 2009.

[46] ISO/IEC/IEEE International Standard 29148 - Systems and Software Engineering - Life Cycle Processes - Requirements Engineering, ISO/IEC/IEEE, 2011.

[47] Klaus Pohl and C. Rupp, Requirements Engineering Fundamentals, 1st ed. Rocky Nook, 2011.

[48] M. Sabou, "An Introduction to Semantic Web Technologies," in *Semantic Web Technologies for Intelligent Engineering Applications*, S. Biffl and M. Sabou Eds. Cham: Springer International Publishing, 2016, pp. 53-81.

[49] "JSC 29948B, ISS IBM THINKPAD SERIES A31P LAPTOP HARDWARE PROJECT TECHNICAL REQUIREMENTS SPECIFICATION." EverySpec. http://everyspec.com/NASA/NASA-JSC/NASA-JSC-PUBS/JSC-29948B_29701/ (accessed February 08, 2022).

[50] M. A. Musen, "The protégé project: a look back and a look forward," *AI Matters,* vol. 1, no. 4, pp. 4-12, doi: 10.1145/2757001.2757003.

[51] "NetworkX - Network Analysis in Python." https://networkx.org (accessed March 7, 2022).