Semantic Parsing of Textual Requirements

Ole Magnus Holter

Department of Informatics, University of Oslo, Oslo, Norway
olemholt@ifi.uio.no

Abstract. Requirements are critical components in the industry, describing qualities that a product or a service needs to have. Most requirements are only available as natural language text embedded in a document. Working with textual requirements is getting increasingly difficult due to the growing number of requirements, and having the requirements available as structured data would be beneficial. However, the work required for the translation of natural language requirements into structured data is daunting. Thus, we need tools to aid in this process. In this Ph.D. project, we propose to use state-of-the-art knowledge extraction techniques and develop novel methods to identify the terms and relationships in a requirement and align them with an existing domain-ontology. To achieve this goal, we must overcome the difficulties in working with both domain-specific technical corpora and ontologies. Furthermore, existing tools and NLP models must be adapted to the domain.

Keywords: Semantic parsing · NLP · RDF · Requirements

1 Introduction

Requirements describe the features and qualities that a product or a service needs to have, including legal regulations. Being essential to most industries today, the requirements usually form part of the legal agreements between parties. Requirements are also used to direct work processes, ensure worker safety, and to reduce environmental impact. In most cases, the requirements are available only within textual documents (e.g., PDF, Word). In large companies, this does not scale well. Moreover, natural language is inherently ambiguous and imprecise; consequently, misunderstandings are common. Besides, the use of natural language documents makes it hard to organize the requirements in a way that avoids requirements to be repeated. Thus, the situation today is that many requirements are hard to find and duplicated or conflicting requirements are not uncommon.

Current solutions for digital management of requirements (e.g., Polarion [22]) focus on better organization of existing natural language requirements. By ensuring that every requirement has a unique identifier across all documents, and by adding metadata, such as about the author of a requirement and comments, single requirements can be uniquely identified in the entire workflow, and changes can be managed for each requirement. Although the decoupling with the document

is an important step, it does not solve the industry's challenges with managing requirements.

An attempt to improve the quality of natural language requirements is to define clear guidelines for writing requirements, for example, the guides for writing requirements by the International Council on Systems Engineering (INCOSE) [11].

The READI project [23] creates standards for requirement modeling and for expressing requirements as structured data. This is a top-down approach to requirement modeling where existing requirements are currently ignored, and the aim is to develop new approaches for describing and modeling. As part of READI, research on how to effectively model requirements in OWL 2 is also being done [14].

Having the requirements described as structured data can open up for novel ways to organize, process, and think about requirements. It could potentially transform how the industry works with requirements. However, even if requirements are not completely modeled, annotation and categorization can prove a useful step towards better requirement management and adherence. By exploiting the hierarchies in taxonomies and by the use of automatic reasoning, identification and maintenance of the requirements will be more manageable, and the identification of duplicate and conflicting requirements can be enabled. In the future, documentation can be automatically generated and sent to the stakeholders. It might even be possible to build applications that automatically retrieve requirements relevant for a project and check automatically whether the requirements are fulfilled.

We cannot, however, ignore the existing textual requirements. The industry is committed to following the existing corpus of textual requirements, and the textual requirements will continue to play an essential role in communication between parties. The existing situation could be improved by having existing requirements translated into a structural representation. However, the cost of manually translating requirements is daunting. Consequently, there is a need for (semi-)automatic tools that can aid the task. Knowledge extraction from general text is hard. Our task, however, is not to understand general text but rather text from the domain of technical requirements, in which we expect the authors to have some degree of adherence to guidelines and aim to be clear and concise.

Further, we expect that existing tools, being trained on general corpus text, are insufficient for this purpose. We also expect the documents to contain non-textual elements (e.g., graphs and tables, which can only be understood in the current context). While these elements are central to the understanding of the requirements, we choose to ignore them and focus only on the text to limit the scope of this project.

The ideal solution to address the industry's challenges with requirements would be a fully automatic system that translates from natural language representation into high-quality structural representation (i.e., an RDF graph). Such a system may not be realistic due to the nature of natural language text being both inherently ambiguous and complex. Even human experts will not agree on how to perform certain translations. We expect, however, that the work towards the

vision of a fully automated system will provide several sub-tasks with a lower level of complexity that could equally benefit requirements management and adherence to requirements in the industry such as the identification of single requirements in the text, the categorization of requirements, and the identification of domain-specific terminology.

The rest of this paper is organized as follows. Section 2 summarizes related work, and Section 3 describes the task in detail. Sections 4 and 5 describe the research methodology and plan for evaluation, respectively, and some preliminary results are presented in Section 6 before the conclusions are presented in Section 7.

2 State of the Art

This Ph.D. project is related to NLP work on industry requirements. Most of the work in this area, however, is related to the field of software development. Winkler and Vogelsang used word vectors and Convolutional Neural Networks (CNN) to identify requirements in a document [26], while Abualhaija et al. propose to use various parsing strategies together with a random forest classifier for the same task [1]. In [24], Sultanov and Hayes propose to use reinforcement learning for requirement traceability. Other works aim at helping authors to express requirements with higher quality [21, 25] and to identify non-functional requirements [3].

While the research in the software industry is relevant to other domains, we cannot assume it to be directly transferable. The challenges in industries such as, for instance, oil and gas, can be quite different from the challenges in the software industry. For example, a major challenge in the software industry is problems of understanding due to limited domain knowledge of software developers and the limited knowledge about software development by the stakeholders [4]. We expect this challenge to be less pronounced in other industries as requirements are often written by professionals in that particular domain.

The Ph.D. project is also related to knowledge extraction in general, machine-reading, and open information extraction. Extracting knowledge from text is traditionally realized as a pipeline where one first extracts named entities before extracting the relations using either handcrafted rules or via supervised learning. The entities and relations are disambiguated and made available in a machine-understandable form. Typically, these tasks require large corpora of manually labeled sentences. Etzioni et al. argue that it is "time for the AI community to set its sights on Machine Reading" [7]. Central to Machine Reading is Open Information Extraction (OpenIE), a paradigm that has a focus on domain independence and unsupervised understanding of text [2].

An important step in the knowledge extraction pipeline is named entity recognition [10]. NER is commonly seen as a sequence labeling task. Rule-based approaches, probabilistic models (e.g., Markov models), and more advanced neural network algorithms are used for this task [15].

Works on entity disambiguation and the detection of emerging entities are also relevant for the Ph.D. project, but are out of the scope of this paper.

Identification of domain-specific terms in domain-specific documents, or automatic terminology acquisition (ATA), is an essential step in many NLP tasks dealing with domain-specific documents and has been studied extensively. Some examples are [6, 12, 13, 19]. TermoStat [6] uses a general domain corpus and identifies (simple and complex) domain-specific terms in an input document by comparing the frequency of the terms between the general domain corpus and the input document. More recent approaches to domain-specific term extraction also use supervised and unsupervised machine learning approaches [13].

Gangemi et al., in the work on FRED [8], propose that natural language can be automatically translated to linked data using classical NLP techniques together with Discourse Representation Theory by first using Semantic Role Labeling (SRL) and NER. The text is then transformed into Discourse Representation Structures (DRTs), which are translated into RDF and OWL 2 statements.

Besides the work on NLP, there is also related work on modeling requirements. The work by Klüwer et al. [14] suggests a model where a requirement is an individual of the class `requirement`. A requirement has a relationship `positedBy` to an individual and a `hasSCDclause` relationship to a clause with the following three properties: *(i)* `hasScope` which is the scope of the requirement (e.g., a *Shell boiler*), *(ii)* `hasCondition` which is an optional condition (e.g., *with a diameter of 1400 mm or greater*), and *(iii)* `hasDemand` what is required (e.g., a *Manhole*). This representation of requirements uses the punning feature of OWL 2 (i.e., it treats classes as individuals).

3 Problem Statement

Our goal is to automatically translate industry requirements into high-quality machine-understandable structured data. For this specific task, quality must be measured both in terms of completeness and correctness. We also want to make the translation conform to a domain-specific ontology. The identification of requirements sentences in the document is in itself a task that is important to the Ph.D. project but is not discussed further in this paper. Assuming that we have correctly identified an individual requirement in a document, we break down the goal into four sub-tasks with increasing complexity. First, we need to identify its main components, namely the scope, the condition (if any), and the demand. The second task is to link these fragments with the relevant classes and properties from a knowledge base. At this point, the approach may also suggest new classes and properties be added to the ontology. The third task is to formalize the relationship as an RDF graph.

Consider the requirement 1.1.5 from DNV GL's "Rules for classification Ships, part 4 – Systems and components, Chapter 7 – Pressure equipment" [5]:

> *1.1.5 Shell boilers with a shell diameter of 1400 mm or greater shall be designed to permit entry of a person and shall be provided with a manhole for this purpose.*

Using the ontology proposed by Klüwer et al. [14], this requirement can be translated into the following RDF graph (in Turtle syntax):

```
ex:1.1.5 a ex:Requirement ;
    ex:hasSCDclause ex:scd1 ;

ex:scd1 a ex:SCDclause ;
    ex:hasScope ex:ShellBoiler ;
    ex:hasCondition ex:cond1 ;
    ex:demandStatement "permit entry of a person" .

ex:cond1 a ex:Condition ;
    ex:subject ex:ShellDiameter ;
    ex:predicate xsd:minInclusive ;
    ex:object 1400 ;
    ex:unit "mm" .

ex:1.1.5b a ex:Requirement ;
    ex:hasSCDclause ex:scd2 .

ex:scd2 a ex:SCDclause ;
    ex:hasScope ex:ShellBoiler ;
    ex:hasCondition ex:cond1 ;
    ex:hasDemand ex:Manhole .
```

Having all the information parsed and resolved against classes and properties is ideal. However, complex statements can be hard to parse and align with an ontology. If we are not able to resolve natural language strings with relevant concepts from the ontology, but instead only label parts of a sentence as scope, condition and demand, then that would already be helpful for the organization of requirements and the retrieval of relevant requirements for a given project, especially in a semi-automatic process with a human in the loop.

From the outlined goal, we formulate the following four research questions for the Ph.D. project.

RQ 1: *To what extent can we automatically translate textual requirements into high-quality machine-understandable structured data?*

We will look at approaches on how to automatically generate RDF graphs from given requirements.

RQ 2: *To what extent can we make the automatic translation conform to a given domain-specific ontology?*

A translation from a textual requirement to structured data is not very helpful if it cannot be used together with existing systems and knowledge bases. By creating a graph that conforms to a domain ontology, however, we can integrate the requirements with other existing requirements and can make effective use of them.

RQ 3: *To what extent can a domain ontology help in processing natural language by providing more accurate parses of textual requirements into a structured representation?*

As domain ontologies describe concepts and relations between concepts in a given domain, they contain useful information that could improve parsing. We will investigate to what extent domain ontologies can help in this step of the process as well.

RQ 4: *Does an automatic translation from textual requirements to a preliminary structured representation, followed by manual improvements, reduce the total time required to produce high-quality structured representations?*

With this question, we want to find out if we can, by using the automatically translated textual requirements, reduce the effort over a manual translation, including potential manual corrections.

Proposed Methods

The translation of the requirements into an RDF graph can be considered a pipeline of smaller tasks. There are many strategies we can use that can give us valuable features that might help to classify and extract knowledge from the documents. These strategies include *(i)* automatic extraction of domain-specific terms. *(ii)* sentence tokenizing and word tokenizing, *(iii)* normalizing words (e.g., lemmatization, case normalizing), *(iv)* POS-tagging, *(v)* chunking (NP chunking), *(vi)* constituency parsing, *(vii)* dependency parsing, *(viii)* Semantic Role Labeling, *(ix)* class recognition (in contrast to NER where individuals are recognized), *(x)* identification of patterns in text, *(xi)* relation extraction, and *(xii)* linking classes and relations to a domain ontology.

Currently, most state-of-the-art systems for these types of processing are using end-to-end neural modeling [27] [28]. One key difficulty for this project, however, is the limited amount of data existing for the domain, making the use of neural modeling challenging. We need to evaluate if such systems, together with weak supervision [18] and transfer learning methods such as in [16], can be used effectively for this task. We may also need to approach the problem using declarative strategies or hybrid strategies.

In requirement texts, we do not expect to find named entities, but more abstract domain-specific terms (T-box terms). Tagging concepts with domain-specific terminology using a few general classes can prove to be a useful feature for the retrieval of knowledge. From the example in Section 3, we would consider `Shell boiler` to be a class (i.e., it does not refer to a specific instantiation of the concept) that is a subclass of `boiler`, which is again a subclass of `container`. This, we believe, can be done, for example, as shown in [17] or by terminology lookup, as proposed in [20].

Relation extraction can also be done using either neural, declarative or hybrid methods. Once some relationships are found, these can be used to find even more concepts and relationships.

The identification of the three elements scope, condition, and demand in a sentence can be thought of as a sequence labeling task. This task is, however, specific to the work on textual requirements, so any training data would have to be created from scratch.

4 Research Methodology

Finding and reading relevant literature is essential as this Ph.D. project requires competence and in-depth knowledge of the state-of-the-art in several domains.

For **RQ 1**, we will test several approaches for the extraction of knowledge from requirements documents and evaluate which approaches are effective for real industry requirements. Further, we will extend existing approaches and devise a new method for knowledge extraction from industry requirements.

We have to develop a quality criteria in order to determine if a translation is of high quality. To evaluate the quality of our method, we have the opportunity to work together with experts both in technical domains and in the general domain of requirements.

For **RQ 2** and **RQ 3**, we will test approaches taken by other methods that deal with linking and annotating textual data and elaborate on these. We have access to technical experts in several domains. However, we have yet to decide on a specific domain where a (possibly incomplete) ontology already exists. We can also make use of industry taxonomies and other available ontologies.

For **RQ 4**, we would need to divide domain experts randomly into two teams where one team does the manual translation of the requirements, and the other team uses the method that we aim to develop during the Ph.D. project. We measure the time and the quality of the translations. We also plan to let the first group do the translations with the system afterward and do qualitative interviews with the domain experts to evaluate the experiences.

5 Evaluation Plan

We will evaluate the method on real industry requirements with the help of domain experts. We intend to manually annotate a set of requirements and let domain experts create translations into structured representations before we agree upon a gold standard. We will also consider the differences between translations and determine a human standard deviation. The gold standard will be shared with the community.

We will evaluate the method in a stacked approach. First, we can evaluate the scope, condition, demand labeling approach, then the linking of the concepts to classes and properties before we evaluate the actual translation into an RDF graph. For each step, we evaluate how suitable existing tools are to solve the problems and then how much our novel approach can improve on the existing tools.

We plan to evaluate the different stages using standard metrics such as accuracy, precision, and recall. We need to define completeness and accuracy of the translation and expected human performance must be defined. Structural differences that are functionally equivalent should be considered equal. Another measure of quality for the actual translation is how good we can translate the individual requirements. (i.e., To what extent are we dealing with natural language, and to what extent do we have classes in the resulting RDF graph).

6 Preliminary results

As an initial experiment, we investigate what can be expected by existing systems on typical textual requirements from the oil and gas industry.

TermoStat [6] identifies more than 1000 domain-specific terms from the DNV GL's requirements for ship classification [5].

AllenNLP [9] is a deep learning library for NLP that includes pretrained models for several common NLP tasks. It comes with an online demo-version that was used to generate the figures 1 and 2, showing quite promising results for the example.

We have also manually annotated requirements from the DNV GL Ship classification document[1] [5]. What we find is that, in most cases, manually identifying the overall scope and demand parts of a single requirement is not difficult. It is, however, challenging to distinguish between a condition and a refinement of the scope (e.g., if it is a subclass of the scope or a condition on the requirement). Some times, the scope is implicit from the structure of the document. We also find that some requirements contain multiple scopes or multiple demands. When identified, some scopes, conditions, and demands are very complex and will not align easily with a taxonomy.

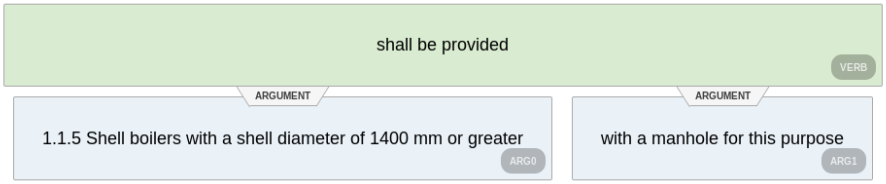

Fig. 1. Open Information Extraction (AllenNLP)

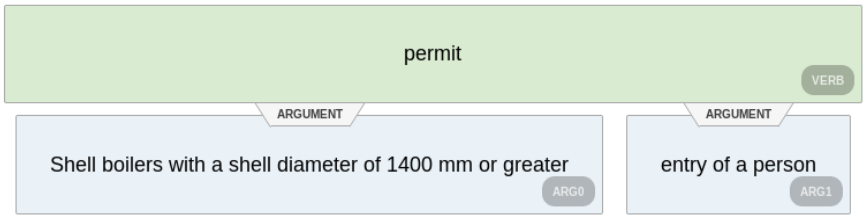

Fig. 2. Semantic Role Labeling (AllenNLP)

[1] The annotation is available at https://gitlab.com/oholter/scd-annotations

7 Conclusions

This Ph.D. project proposes to translate natural language industry requirements to structured data automatically, by the use state-of-the-art knowledge extraction techniques This is, however, not trivial as the techniques mostly depend on large amount of training data, and because natural language is complex and ambiguous.

The translation can be done with different levels of complexity, all of which could be of interest to the industry. First, identify the three main components of a single requirement from the text, namely scope, condition, and demand. Second, to link these fragments to relevant classes and properties from a knowledge base. Third, to formalize the relationship as an RDF graph.

Acknowledgements

The Ph.D. project is supervised by Basil Ell, Martin Giese, and Lilja Øvrelid and is funded by the SIRIUS centre[2]: Norwegian Research Council project number 237898. It is co-funded by partner companies, including DNV GL and Equinor.

References

1. Abualhaija, S., Arora, C., et al.: A Machine Learning-Based Approach for Demarcating Requirements in Textual Specifications. pp. 51–62. RE (2019)
2. Betteridge, J., Carlson, A., et al.: Toward Never Ending Language Learning. In: AAAI spring symposium: Learning by reading and learning to read. pp. 1–2 (2009)
3. Casamayor, A., Godoy, D., Campo, M.: Identification of Non-Functional Requirements in Textual Specifications: A Semi-Supervised Learning Approach. Information and Software Technology **52**(4), 436–445 (2010)
4. Christel, M.G., Kang, K.C.: Issues in Requirements Elicitation. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst (1992)
5. DNV GL: DNV-RU-SHIP-Pt4-Ch7: Rules for classification - Ships, https://rules.dnvgl.com/docs/pdf/DNVGL/RU-SHIP/2018-01/DNVGL-RU-SHIP-Pt4Ch7.pdf, accessed: 2020-02-07
6. Drouin, P.: Term Extraction Using Non-Technical Corpora as a Point of Leverage. Terminology **9**(1), 99–115 (2003)
7. Etzioni, O., Banko, M., Cafarella, M.J.: Machine Reading. In: AAAI. vol. 6, pp. 1517–1519 (2006)
8. Gangemi, A., Presutti, V., Recupero, R., et al.: Semantic Web Machine Reading with FRED. Semantic Web **8**(6), 873–893 (2017)
9. Gardner, M., Grus, J., et al.: AllenNLP: A Deep Semantic Natural Language Processing Platform. arXiv:1803.07640 (2018)
10. Grishman, R., Sundheim, B.: Message Understanding Conference-6: A Brief History. p. 6. COLING (1996)
11. International Council on Systems Engineering: INCOSE, https://www.incose.org/, accessed: 2020-01-21

[2] http://sirius-labs.no

12. Jacquemin, C., Bourigault, D.: Term Extraction and Automatic Indexing, vol. 1. Oxford University Press (2012)
13. Judea, A., Schutze, H., Bruegmann, S.: Unsupervised Training Set Generation for Automatic Acquisition of Technical Terminology in Patents. p. 11. COLING (2014)
14. Klüwer, J.W., Waaler, A.: Reified requirements ontology. (2019), https://w3id.org/requirement-ontology/ontology/core/A01A, accessed: 2020-01-30
15. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural Architectures for Named Entity Recognition. arXiv:1603.01360 (2016)
16. Lee, J.Y., Dernoncourt, F., et al.: Transfer Learning for Named-Entity Recognition with Neural Networks. arXiv:1705.06273 (2017)
17. Nooralahzadeh, F., Lønning, J.T., Øvrelid, L.: Reinforcement-Based Denoising of Distantly Supervised NER with Partial Annotation. pp. 225–233. DeepLo 2019
18. Ratner, A., De Sa, C., Wu, S., Selsam, D., Ré, C.: Data Programming: Creating Large Training Sets, Quickly. NeurIPS (2016)
19. Rigouts Terryn, A., Drouin, P., Hoste, V., Lefever, E.: Analysing the Impact of Supervised Machine Learning on Automatic Term Extraction: HAMLET vs TermoStat. pp. 1012–1021. RANLP 2019
20. Savova, G.K., Masanz, J.J., et al.: Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, component evaluation and applications. J Am Med Inform Assoc (2010)
21. Seresht, S.M., Ormandjieva, O., Sabra, S.: Automatic Conceptual Analysis of User Requirements with the Requirements Engineering Assistance Diagnostic (READ) Tool. pp. 133–142. SERA 2008
22. Siemens AG: Polarion ALM, https://polarion.plm.automation.siemens.com, accessed: 2020-01-07
23. SIRIUS: DREAM and READI: Cooperation to Manage Digital Requirements, https://sirius-labs.no/dream-and-readi-cooperation-to-manage-digital-requirements/, accessed: 2019-10-15
24. Sultanov, H., Hayes, J.H.: Application of Reinforcement Learning to Requirements Engineering: Requirements Tracing. pp. 52–61. RE 2013
25. Wang, Y.: Automatic Semantic Analysis of Software Requirements ThroughMachine Learning and Ontology Approach. Journal of Shanghai Jiaotong University (Science) **21**(6), 692–701 (2016)
26. Winkler, J., Vogelsang, A.: Automatic Classification of Requirements Based on Convolutional Neural Networks. pp. 39–45. REW 2016
27. Wu, S., He, Y.: Enriching Pre-trained Language Model with Entity Information for Relation Classification. eprint arXiv:1905.08284 (2019)
28. Zhao, Y., Wan, H., Gao, J., Lin, Y.: Improving relation classification by entity pair graph. In: Asian Conference on Machine Learning. pp. 1156–1171 (2019)