Towards a Corpus of Requirements Documents Enriched with Semantic Frame Annotations

[Link to publication record in Manchester Research Explorer](#)

Citation for published version (APA):

Alhoshan, W., & Batista-Navarro, R. T. (2018). *Towards a Corpus of Requirements Documents Enriched with Semantic Frame Annotations*.

Citing this paper

Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights

Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy

If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [<http://man.ac.uk/04Y6Bo>] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.

Towards a Corpus of Requirements Documents Enriched with Semantic Frame Annotations

Waad Alhoshan, Riza Batista-Navarro and Liping Zhao
School of Computer Science, University of Manchester
Oxford Road, Manchester M13 9PL, UK

Abstract—Software requirements are typically written in natural language, which need to be transformed into a more formal representation. Natural language processing techniques have been applied to aid in this transformation. Semantic parsing, for instance, adds semantic structure to text. It however requires supporting corpora which are still missing in requirements engineering. To address this gap, we developed FN-RE, a corpus of requirements documents, which was annotated based on semantic frames in FrameNet. Each requirement statement was manually labelled by two annotators by selecting suitable semantic frames and related frame elements. We obtained an average agreement of 72.85% between the two annotators, measured by F-score, thus indicating that the annotations provided in our corpus are reliable.

Index Terms— Requirements Documents, Requirements Corpus, Semantic Annotation, Semantic Frames, FrameNet

I. INTRODUCTION

Requirements engineering (RE) is concerned with understanding user requirements and transforming them into a form that is amenable for software design and implementation [1]. However, user requirements, especially at the early stages of RE, are usually written in natural language (NL) [2]. Consequently, the semantics of requirements become obscured, preventing them from being readily searchable or understandable. This barrier can be potentially overcome with the use of natural language processing (NLP) techniques, which require annotated corpora supporting their development and evaluation [2].

In this paper, we aim to foster the development of NLP-based methods for semantic parsing of requirements. To this end, we constructed a corpus of requirements documents annotated with semantic frame information from FrameNet [3]. Containing 220 requirement statements collected from various online sources, the corpus labelled with semantic frame annotations was stored in an XML-based format and is publicly available at <https://goo.gl/R4soVQ>.

The rest of this paper is organised as follows. Section II gives a brief overview of semantic frames and FrameNet. Section III describes our FrameNet-based annotation approach. This is then followed by Section IV where we describe the resulting FN-RE corpus. Section V provides a discussion of the applications and limitations of the corpus. Finally, Section VI summarises our contributions and outlines our plans for future work.

II. OVERVIEW OF SEMANTIC FRAMES AND FRAMENET

A *semantic frame* is a coherent structured representation that encapsulates associations between related concepts in a

statement [4]. Based upon the theory of frame semantics [5], a semantic frame allows for the full understanding of the meaning of a word (e.g., an action signified by a verb) by providing access to related essential knowledge (e.g., participants in the action such as the "who" and "what").

More than 1, 200 semantic frames have been catalogued in detail in FrameNet¹, a web-based general-domain computational lexicon [3]. For every semantic frame in FrameNet (FN), the following information is provided: frame title, definition, frame elements and lexical units (LUs). LUs are words that evoke the frame, represented as a combination of their lemmatised form and part-of-speech (POS) tag. The concept of creation, for example, which is encoded in FN as a semantic frame entitled *Creating*, can be evoked by the LUs *create.v* and *generate.v* (where *v* stands for verb). Its frame elements include *Creator*, *Created_entity* and *Cause*, among many others.

In this work, we propose to apply such frames on natural language requirements, in order to represent their semantics in a structured manner. Figure 1 shows a requirement statement whose meaning had been annotated based on semantic frame information from FrameNet.

Fig 1. Examples of semantic frames (and their associated frame elements) in FrameNet applied to a requirement statement.

¹ <https://framenet.icsi.berkeley.edu/>

As illustrated in the example in Figure 1, multiple semantic frames might be applicable to the same statement. Apart from the *Creating* frame, the following two other semantic frames were also evoked:

- *Having_or_lacking_access*. This semantic frame describes access to a location or resource despite a potential barrier. The LU that evokes this frame in this example is the verb *access* (represented in FN as *access.v*).
- *Conditional_occurrence*. This semantic frame describes the possibility of an event occurring, depending on whether a certain situation holds. In this example, the frame is evoked by the LU *if* (represented in FN as *if.scon* where *scon* stands for subordinating conjunction).

Furthermore, the noun *system* can also evoke a *Gizmo* frame to denote a piece of equipment used for a purpose.

By annotating requirements documents based on semantic frame information from FrameNet in this manner, we seek to make the meaning within requirement statements structured and explicit. The resulting annotated corpus can then support the development and evaluation of NLP-based methods for semantic parsing of text describing requirements.

III. FRAMENET-BASED ANNOTATION

In this section, we present details of our FrameNet-based annotation approach which consists of the four steps outlined below.

A. Selection and Pre-processing of Documents

As a preliminary step towards collecting a representative set of requirements documents, we formed a Google search query containing keywords such as “software description”, “natural language requirements” and “software requirements specification”. Additionally, we employed snowball sampling and found additional requirements from various sources such as web blogs, research articles (together with their corresponding datasets), lecture materials, and industrial or commercial documents. This step resulted in the collection of 34 requirements documents varying in length. The NLTK tool for sentence boundary detection² was then applied on the 34 documents. After manually verifying the sentence boundary detection results, a total of 1,148 sentences were obtained (corresponding to 21,012 tokens). These requirements documents were then encoded in an XML-based format following the scheme presented in [6].

B. Frame Selection

This step is concerned with processing requirement statements to label them with the semantic frames they evoke. Firstly, we developed a method for automatically matching words in the statements against LUs in FrameNet. The tokens contained in requirement statements were lemmatised and assigned part-of-speech (POS) tags using NLTK in python. We then attempted to match each token (together with its

lemma and POS tag) against LUs in FN, via the FN application programming interface (API) available in NLTK³.

Afterwards, the evoked candidate semantic frames were presented to two annotators for manual validation, i.e., selecting the frames that best capture the meaning expressed in a given statement. The first annotator (Annotator A) is a requirements engineer with five years of experience in the IT industry. The second annotator (Annotator B) is the first author of this paper whose PhD project focusses on the use of NLP techniques to support RE tasks. A procedure for harmonising their selections was carried out manually to minimise inconsistencies. The final result of this step consists of requirement statements labelled with the FN semantic frames that were evoked by matched LUs. For further details on the manual validation and harmonisation of evoked frames, we refer the reader to our previous work [7].

C. Frame Element Annotation

On top of the semantic frames in FN evoked in each requirement statement, associated frame elements were also annotated. We cast this step as a *semantic role labelling* task, i.e., the assignment of roles to words or word sequences in a statement [8]. In this task, the arguments of a given predicate (e.g., a verb signifying an action) are identified and assigned corresponding roles, e.g., *agent*, *experiencer*, *instrument*. In our case, matched LUs that evoke frames (from the previous step) are treated as predicates. The goal of this task is to then identify respective frame elements by marking up the arguments of these predicates and assigning them roles. To this end, we made use of SALTO, a Java-based annotation tool specifically developed for the annotation of semantic frames [9]. Its graphical interface supports the visualisation as well as drag-and-drop editing of relationships between predicates and arguments.

We firstly encoded our requirements documents—with partial frame annotations from the previous step (i.e., matched LUs and evoked FN frames)—in the format required by SALTO, i.e., SALSA/TIGER XML [10]. Additionally, an XML-formatted file containing information on FN frame elements (obtained via the FN API) was provided to the tool as a lookup list. For each requirement statement, the tool presents to the annotator: (1) the tokens contained in the text, (2) matched LUs and the FN frames they evoke, and (3) the frame elements to choose from according to the FN lookup list. The annotator then revises each evoked frame by carefully reading its definition and the accompanying annotation examples from FrameNet. Afterwards, SALTO is used to select the desired elements for each frame by attaching a related token (or sequence of tokens) to the predicate and specifying its semantic role.

Figure 2 presents an example of a requirement statement that evokes the *Creating* frame which has 19 associated frame elements⁴ according to FrameNet. Out of these, the five frame elements that apply to the statement were identified with the aid of SALTO. Each of these elements were then linked to the

² <http://www.nltk.org>

³ <http://www.nltk.org/howto/framenet.html>

⁴ framenet2.icsi.berkeley.edu/fnReports/data/frameIndex.xml?frame=Creating

corresponding token or token sequence, e.g., the noun “system” for *Creator* and the noun phrase (NP) “records of user activities” for *Created_entity*.

As our annotators were carrying out this step, we found it necessary to revisit results of the previous step, i.e., the semantic frames which were manually selected for each requirement statement. This is because in some cases, a closer look at candidate frame elements provided our annotators with a different understanding of the evoked frames that were selected previously. As a consequence, some frames were additionally selected, some were replaced by other frames, while others were discarded. In the case of discarded frames, the corresponding LUs were documented to allow us to analyse (as part of future work) whether any frames or LUs are currently missing in FN. The annotations resulting from this step were then stored by SALTO in the SALSA/TIGER XML format.

This step proved to be much more time-consuming than the previous one. While we were aiming to annotate frame elements in all of the 34 requirements documents in our collection, thus far we have completed this step for only 18 of them (containing a total of 220 requirement statements).

D. Annotation Validation and Harmonisation

In order to assess the consistency of annotations between our two annotators, we measured the agreement between them, (i.e., inter-annotator agreement) by calculating the value of F-score: the harmonic mean of recall and precision, as shown in Equation 1.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad \text{where,} \quad \text{Eq. 1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \text{ and } \text{Recall} = \frac{TP}{TP + FN}$$

To calculate the values of precision and recall, we counted the number of true positives (TPs), false positives (FPs) and false negatives (FNs) obtained by Annotator A with respect to Annotator B's annotations, which—for the purposes of F-score calculation—were treated as gold standard. TPs are the set of Annotator B's annotations that overlap with Annotator A's annotations, while those which were missed correspond to FNs. Meanwhile, FPs are comprised of the annotations from

Annotator A which are not in Annotator B's annotations (and hence are not in the gold standard).

Inter-annotator agreement (IAA) was measured over our 18 requirements documents, at three different levels:

- *Evoked frames*: the extent to which both annotators selected the same frames.
- *Frame elements (roles only)*: for frames that the two annotators agree on, the extent to which they agree on frame elements considering only roles.
- *Frame elements (roles and text spans)*: for frames and frame element roles that the two annotators agree on, the extent to which they agree on frame elements if text spans are considered.

Disagreements between the two annotators were resolved through a procedure for annotation harmonisation. That is, for every discrepancy at any of the three levels described above, Annotator B revisited her annotations, this time taking into consideration Annotator A's annotations. In cases where she was convinced that Annotator A's perspective was more correct, she modified her annotations; otherwise, she kept her original annotations. Annotations resulting from the harmonisation procedure formed the basis of our FN-RE corpus of requirements documents.

IV. CORPUS DESCRIPTION

Our FN-RE corpus consists of 18 frame-annotated requirements documents. The FN semantic frame annotations were stored separately from the requirements documents. While the documents were stored following an extended version of the XML schema proposed by [6], the semantic frame annotations were converted from SALSA/TIGER XML to the FrameNet XML format [3]. The corpus is publicly available at <https://goo.gl/R4soVQ>.

The results of inter-annotator agreement calculation are presented in Table 1. Based on these, we can say that for each

Annotation Level	IAA (F-score)
Evoked frames	78.50%
Frame elements (roles only)	76.90%
Frame elements (roles and text spans)	72.85%

Table 1. Inter-annotator agreement scores for each annotation level.

of the three annotation levels considered, satisfactory agreement between our annotators was obtained. Hence, we can consider the annotations in our FN-RE corpus as reliable.

Upon performing frequency analysis over the final set of annotations, the results presented in Table 2 were obtained.

Frame	Freq	Frame Elements	Freq
1. <i>Sending</i> Example: <i>Every month the customers will be sent a statement by email, together with a list of special offers.</i>	41	Theme	13
		Sender	8
		Goal	5
		Recipient	4
		Explanation	3
		Cotheme	2
		Time	2
		Transport means	2
		Purpose	1
		Source	1
2. <i>Conditional_occurrence</i> Example: shown in Fig 1	38	Profiled possibility	20
		Consequence	18
3. <i>Inclusion</i> Example: <i>The system has an option for shoppers to register with the shop.</i>	37	Part	17
		Total	17
		Explanation	2
		Purpose	1

Table 2. The three most frequently evoked FN frames in the FN-RE corpus, together with the corresponding most frequently assigned frame elements.

V. APPLICATIONS & LIMITATIONS

Although a number of FN frame-annotated corpora exist [3], [11], [12], none so far have focussed on the annotation of requirements documents. Our FN-RE corpus was developed to address this gap, with the specific aim to facilitate the development and evaluation of semantic parsing tools for requirement statements. By automatically labelling requirement statements with the FN frames they evoke as well as their corresponding frame elements; such tools can put the meaning of these statements in a structured and explicit form. This in turn will support a number of analytical tasks crucial to RE. Examples include enhanced *search* as well as *reuse* of requirement statements through similarity detection based on the frames (and frame elements) that they were automatically labelled with (as shown in [13]). Furthermore, in the way of supporting requirements *modelling*, labels produced by semantic parsing can enable the visualisation of the meaning expressed within requirements documents (as shown in [14]).

However, the corpus—which is currently undergoing expansion—comes with some limitations. With its small size (i.e., 220 requirement statements with 123 unique FN frames), it cannot support the development of supervised machine learning-based methods for semantic parsing. Furthermore, the annotations it currently contains do not account for semantic frames that are missing in FrameNet but are nevertheless valid.

VI. CONCLUSION AND FUTURE WORK

In this paper, we described our work on the development of FN-RE, a corpus of requirements documents annotated with

semantic frame information from FrameNet. Although the corpus is a work-in-progress, we contribute our current results to the community, in order to encourage the development and evaluation of semantic parsing tools for requirement statements.

One of our immediate steps is to extend the corpus by annotating more requirements documents. To accelerate this process, we shall explore the application of state-of-the-art semantic role labelling tools⁵ on requirement statements to automatically generate frame annotations which we will then manually validate. We also plan to conduct an analysis of our requirements documents to identify—and potentially propose—frames that currently might be lacking in FrameNet.

REFERENCES

- [1] J. Dick, E. Hull, and K. Jackson, *Requirements engineering*: Springer, 2017.
- [2] A. Ferrari, F. Dell'Orletta, A. Esuli, V. Gervasi, and S. Gnesi, "Natural Language Requirements Processing: A 4D Vision," *IEEE Software*, vol. 34, pp. 28-35, 2017.
- [3] C. F. Baker, "FrameNet: Frame Semantic Annotation in Practice," in *Handbook of Linguistic Annotation*, ed: Springer, 2017, pp. 771-811.
- [4] M. R. Petrucci, "Frame semantics," *Handbook of pragmatics*, pp. 1-13, 1996.
- [5] C. J. Fillmore, "Frame semantics and the nature of language," *Annals of the New York Academy of Sciences*, vol. 280, pp. 20-32, 1976.
- [6] A. Ferrari, G. O. Spagnolo, and S. Gnesi, "PURE: A Dataset of Public Requirements Documents," in *International Requirements Engineering Conference.*, 2017, pp. 502-505.
- [7] W. Alhoshan, R. Batista-navarro, and L. Zhao, "A FrameNet-based Approach for Annotating Software Requirements," in *International Conference on Language Resources and Evaluation*, 2018.
- [8] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," *Computational linguistics*, vol. 28, pp. 245-288, 2002.
- [9] A. Burchardt, K. Erk, A. Frank, A. Kowalski, S. Pado, and M. Pinkal, "SALTO—a versatile multi-level annotation tool," in *International Conference on Language Resources and Evaluation*, 2006.
- [10] K. Erk and S. Pado, "A Powerful and Versatile XML Format for Representing Role-semantic Annotation," in *International Conference on Language Resources and Evaluation*, 2004.
- [11] A. Burchardt and M. Pennacchiotti, "FATE: a FrameNet-Annotated Corpus for Textual Entailment," in *LREC*, 2008.
- [12] A. Dolbey, M. Ellsworth, and J. Scheffczyk, "BioFrameNet: A Domain-Specific FrameNet Extension with Links to Biomedical Ontologies," in *KR-MED*, 2006.
- [13] M. Bano, A. Ferrari, D. Zowghi, V. Gervasi, and S. Gnesi, "Automated Service Selection Using Natural Language Processing," in *Requirements Engineering in the Big Data Era*, ed: Springer, 2015, pp. 3-17.
- [14] K. J. Letsholo, L. Zhao, and E.-V. Chioasca, "TRAM: A tool for transforming textual requirements into analysis models," presented at the International Conference on Automated Software Engineering, 2013.

⁵ <http://www.cs.cmu.edu/~ark/SEMAFOR/>