Using Frame Embeddings to Identify Semantically Related Software Requirements

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

Abstract

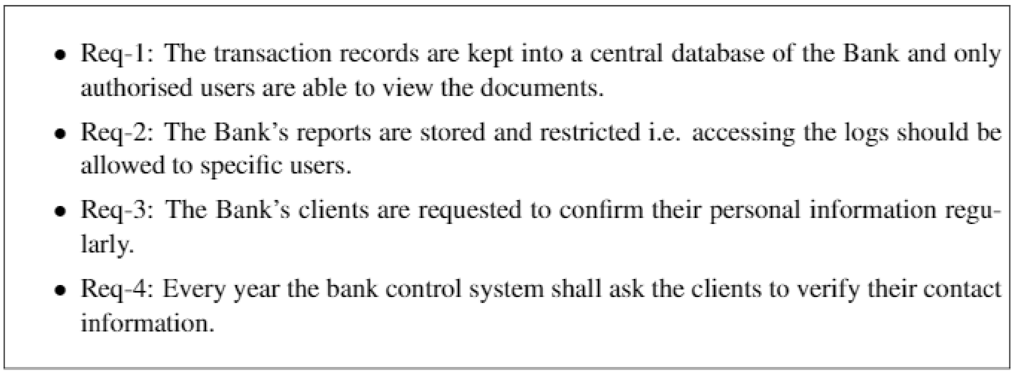
In requirements engineering (RE), software requirements typically come in the form of unstructured text, written in natural language. Consequently, identifying related requirements becomes a time-consuming task. In this paper, we propose a novel method for measuring semantic relatedness between software requirements, with the aim to develop RE tools that can automate the process of detecting traceability links in requirements documents. The proposed method is underpinned by an embedding-based representation of semantic frames in FrameNet, trained on a large corpus of user requirements. Applying the method to the task of detecting semantically related software requirements, the performance of the proposed method was evaluated against a manually labelled corpus and baseline system. Our method obtained satisfactory performance in terms of F-score (86.36%) against a manually labelled data set of software requirements, and outperformed the baseline system by 24 percentage points. These encouraging results demonstrate the potential of our method to be integrated with RE tools for facilitating software requirement analysis and traceability tasks.

1 Introduction

Requirements engineering (RE) is concerned with the elicitation and specification of user requirements for the purposes of software development [HJD17]. In the initial phases of RE, requirements are typically written in natural language [FDE$^+$17], examples of which are shown in Figure 1.

Finding related or similar requirements within a document is an essential task in order to understand linkages between them [HJD17]. Considering the four requirements presented in Figure 1, we can say that both Req-1 and Req-2 are related to file restriction, while Req-3 and Req-4 describe the necessity to update the bank's client data. However, it is a challenging RE task to analyse requirements to find relations between them, either explicit or implicit, because requirements documents are often very long, and written inconsistently by different stakeholders [HJD17]. This problem is due to the semantic ambiguity and incompleteness inherent to natural language [FDE$^+$17]. Moreover, manually analysing requirements is a time-consuming and error-prone procedure [HJD17, GF94, MNX12].

In this paper, detecting semantic relatedness between requirements is defined as the process of finding semantically connected events (e.g., *" The user shall use the software to **send** files."* and *"The user shall **receive** documents as well"*) whereas detecting requirements similarity is defined as the process of finding requirements that discuss similar events (e.g., *" The user shall use the software to **send** files."* and *"The user shall **sub-***



- Req-1: The transaction records are kept into a central database of the Bank and only authorised users are able to view the documents.
- Req-2: The Bank's reports are stored and restricted i.e. accessing the logs should be allowed to specific users.
- Req-3: The Bank's clients are requested to confirm their personal information regularly.
- Req-4: Every year the bank control system shall ask the clients to verify their contact information.

Figure 1: Examples of natural-language software requirements.

***mit** files via the system."*). Detecting such relatedness and similarity in requirements thus requires semantic processing [MNX12].

For instance, semantic relatedness has drawn the attention of RE researchers with the goal of automating the creation of traceability links within requirement documents [MNX12, MN15]. In addition, semantic relatedness constitutes a more intuitive method for tracing requirements, since it provides a holistic view and mimics the human mental model in finding similar or related concepts [MN15]. Tracing such relatedness requires representing the natural language—hence, unstructured—requirements, in a more structured manner. Afterwards, the structured requirements can be analysed for semantic relatedness between them.

Techniques in natural language processing (NLP) offer a viable solution to many tasks in RE [DFFP18], including requirements analysis and traceability [MNX12, MN15]. However, majority of the available NLP techniques and resources are not domain-specific, i.e., they are trained or built on general-domain text (e.g., news articles) [DFFP18]. For this reason, a recent research direction in RE has called for "customizing general NLP techniques to make them applicable for solving the problems requirements engineers face in their daily practice" [DFFP18].

One of the recent trends in NLP research is the use of *word embeddings*, which are a semantic representation of a word that captures the context in which it has been used, within a corpus of documents [MCCD13]. They are learned based on the principle of distributional semantics, which posits that words occurring in the same context have similar or related meanings [Har54]. Deep learning offers a framework for representing word context as real-valued vectors, that goes beyond the counting of co-occurrences and takes into account word order as well [BDVJ03].

In this paper, we present and demonstrate a method for measuring semantic relatedness between software requirements, which is underpinned by embeddings representing semantic frames in FrameNet [FB01, Bak17], trained on a domain-specific corpus using deep learning techniques. In our previous work, we constructed FN-RE, a corpus of software descriptions enriched with semantic frame annotations [ABNZ18a, ABNZ18b]. We also proposed a method for detecting implicit relations between semantic frames, through the measurement of similarity between them [AZBN18]. Specifically, we made use of pre-trained word embedding vectors [ECS18] as a means for representing semantic frames, and then computed their similarity, i.e., the cosine distance between any two FrameNet frames X and Y annotated in the FN-RE corpus. Our initial validation, as reported in [AZBN18], yielded an F-score of 83% against human . In this paper, we extend our previous work to demonstrate the measurement of relatedness—based on semantic frames—at the requirement statement level.

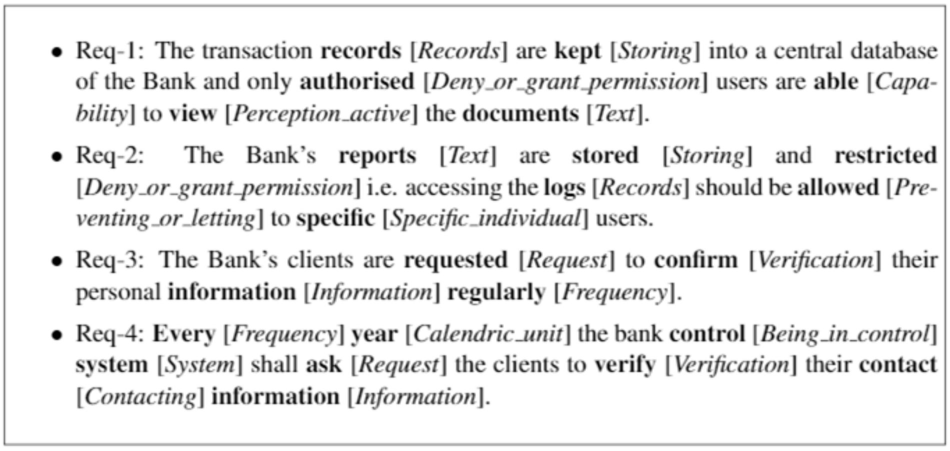
The rest of this paper is organised as follows: Section 2 presents a brief background on FrameNet semantic frames. Section 3 discusses our proposed method for measuring semantic relatedness between software requirements. In Section 4, we explain our evaluation procedures, and then we report results relative to a manually labelled data set and a baseline system. Finally, we conclude and briefly discuss our ongoing work in Section 5.

2 Analysing requirements using FrameNet

A semantic frame is a coherent structured representation that sums up the meanings given by a statement. According to Fillmore's theory, a semantic frame allows for 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 involved in the action such as the "who" and "what") [Fil76].

FrameNet is a computational lexicon[1] implemented based on the semantic frames theory [BFL98, Bak17]. The lexicon, accessible via a web-based interface, holds more than 1,200 semantic frames. For every semantic frame in FrameNet, the following information is given: the frame title, definition, a list of 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. For example, the concept of creation which is included in FrameNet 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 specify the participants involved, namely, the *Creator*, *Created_entity* and *Cause*, among many others.

The meaning of a given piece of text can be represented in a more structured and explicit form, by annotating the semantic frames it contains. Figure 2 shows examples of requirement statements which have been annotated based on semantic frame information in FrameNet. Each LU evokes a semantic frame that conveys a specific



- Req-1: The transaction **records** [*Records*] are **kept** [*Storing*] into a central database of the Bank and only **authorised** [*Deny_or_grant_permission*] users are **able** [*Capability*] to **view** [*Perception_active*] the **documents** [*Text*].
- Req-2: The Bank's **reports** [*Text*] are **stored** [*Storing*] and **restricted** [*Deny_or_grant_permission*] i.e. accessing the **logs** [*Records*] should be **allowed** [*Preventing_or_letting*] to **specific** [*Specific_individual*] users.
- Req-3: The Bank's clients are **requested** [*Request*] to **confirm** [*Verification*] their personal **information** [*Information*] **regularly** [*Frequency*].
- Req-4: **Every** [*Frequency*] **year** [*Calendric_unit*] the bank **control** [*Being_in_control*] **system** [*System*] shall **ask** [*Request*] the clients to **verify** [*Verification*] their **contact** [*Contacting*] **information** [*Information*].

Figure 2: Software requirements annotated based on FrameNet. The associated lexical units appear in bold font and are followed by the titles of the frames that they evoke, denoted in square brackets.

concept. Words such as "authorised" and "restricted", for instance, evoke the *Deny_or_grant_permission* frame in FrameNet, while words such as "confirm", and "verify" evoke the *Verification* frame. It can be observed that although Req-1 and Req-2 both pertain to file restriction, they contain words that are lexically different, e.g., "authorised" vs. "restricted", "records" vs. "logs", "kept" vs. "stored". However, these words evoke the same semantic frames, e.g., "kept" and "stored" both evoke the *Storing* frame.

Tracing the similarity or relatedness between requirements, e.g., between Req-1 and Req-2, or between Req-3 and Req-4, is a non-trivial task. Apart from lexical variation, the syntax structures of these software statements are not identical, thus making the use of heuristic techniques (e.g., word matching) unsuitable. To overcome such issues, there is a need to formulate a method that represents the under-specified meanings of such requirements in a structured form, which can then be used as a basis for automatically measuring relatedness.

3 The Proposed Method

In this section, we present our method for detecting semantic relatedness between software requirements. That is, for any two requirement statements, we measured the semantic relatedness between them on the basis of the FrameNet semantic frames that their lexical units evoke.

3.1 Preparation of Training Data

Whereas our previous work [AZBN18] employed pre-trained word embeddings learned on a corpus of Stack Overflow messages [ECS18], in this work we decided to train our own embeddings on a corpus of documents that are more similar to software requirements, i.e., a collection of user reviews of mobile applications. Using the web-based AppFollow tool[2], reviews from different mobile application repositories (e.g., Apple Store and Google Play) were retrieved. The user reviews covered different categories of mobile applications, i.e., business, sports,

[1]https://framenet.icsi.berkeley.edu
[2]https://appfollow.io

health, travel, technology, security, games, music, photos, videos, shopping, lifestyle, books, social networking, finance. While each review came with metadata such as review date, title and per-user application rating, we took into consideration only the textual content of the reviews. This resulted in a total of 3,019,385 unique reviews/documents in our training data set.

The documents in the training data set were then preprocessed with the following steps: sentence splitting, tokenisation, stop-word removal, part-of-speech (POS) tagging and lemmatisation. The preprocessing results allowed us to automatically check for the occurrence of LUs (associated with semantic frames) catalogued in FrameNet, in order to assess the data set's coverage of semantic frames. Based on this, we were able to determine that our mobile application reviews data set covers all of the 123 semantic frames annotated in the FN-RE corpus[3].

3.2 Training of Word Embeddings

Utilising the preprocessed mobile application reviews data set as a corpus, we trained word embeddings using the continuous bag-of-words (CBOW) learning method of Word2Vec [MCCD13]. A word embedding vector was trained for each LU, which was represented as a combination of its lemmatised form and POS tag. Taking into account the POS tag of an LU makes it possible to train different vectors for words with the same lemma but different parts of speech. It is preferable, for example, to train a vector for "form" as a verb (*form.v*) that is different from the vector for "form" as a noun (*form.n*). The size of each vector was set to 300, following previously reported work [SP18, MCCD13].

3.3 Generation of Frame Embeddings

The word embedding vectors resulting from the previous step were then used to form an embedding-based representation of semantic frames, i.e., *frame embeddings.* That is, for any given semantic frame F, we collected the vectors corresponding to the LUs that evoke it. The average of these LU vectors is then computed and taken as the frame embedding for F. For instance, as 11 LUs are associated with the *Creating* frame in FrameNet, a vector containing the average over the 11 word embedding vectors corresponding to these LUs was obtained as part of this step.

3.4 Measuring Frame-to-Frame Relatedness

The generated frame embeddings were employed in computing relatedness between semantic frames. Following our method described in [AZBN18], we used the cosine similarity metric. For FrameNet frames X and Y, let $R(X,Y)$ denote the relatedness between these two frames:

$$R(X,Y) = cos(X,Y) = \frac{\mathbf{F_X} \cdot \mathbf{F_Y}}{\|\mathbf{F_X}\| \|\mathbf{F_Y}\|} \tag{1}$$

where $\mathbf{F_X}$ and $\mathbf{F_Y}$ are the frame embedding vectors for X and Y, respectively.

The cosine similarity metric measures the angle between two vectors (i.e., frame embeddings). If the vectors are close to parallel (e.g., with $R(X,Y) \approx 1$) then we consider the frames as similar, whereas if the vectors are orthogonal (i.e., with $R(X,Y) \approx 0$), then we can say that the frames are not related.

3.5 Measuring Semantic Relatedness between Requirements

A requirement statement may evoke one or more frames in FrameNet. To measure the semantic relatedness between any two requirement statements A and B (i.e., a requirement pair), we generated a frame-to-frame similarity matrix. The rows of this matrix correspond to the frames evoked in statement A while its columns correspond to the frames evoked in the other statement B. Its cells hold the cosine similarity between two frames F_A and F_B, where F_A is a frame evoked in statement A while F_B is a frame evoked in the other statement B. Given that statement A contains m frames while statement B contains n frames:

$$\begin{aligned} A &= (F_{A1}, F_{A2}, \ldots, F_{Am}) \\ B &= (F_{B1}, F_{B2}, \ldots, F_{Bn}) \end{aligned} \tag{2}$$

[3]https://zenodo.org/record/1291660

The values of the cells of the $n \times m$ matrix M were determined based on the following equation, where i is the ith frame of statement A and j is the jth frame of statement B, and R is a relatedness score based on cosine similarity, as described above:

$$M_{i,j} = R(F_{Ai}, F_{Bj}) \tag{3}$$

With the cells of matrix M populated in the manner just described, we took the average over all of its cells. This resulted in a real-valued score that corresponds to the semantic relatedness between statement A and statement B.

4 Evaluation

For evaluating the proposed method for determining semantic relatedness between requirement statement pairs—which we will henceforth refer to as the F2F (or frame-to-frame) method—we used a data set containing human judgements (or labels) as a gold standard. Furthermore, for comparison, we implemented a baseline system underpinned only by word embeddings.

4.1 Manually Labelled Requirement Statement Pairs

We constructed a new data set, SEM-REQ, in which pairs of requirement statements have been manually assigned labels indicating semantic relatedness. This was drawn from the FN-RE corpus which consists of 18 documents, containing a total of 220 requirement statements. At least three requirement statements were randomly selected from each document in FN-RE, where each statement evokes at least one FrameNet semantic frame. This resulted in a set of 60 requirement statements. Each of these was paired with every other statement in the set, generating a total of 1,770 requirements pairs which were all included in SEM-REQ.

We devised a simple scheme for annotating each pair, whereby "Yes" was assigned to a pair if its two requirements are semantically related, and 'No" otherwise. Each pair in SEM-REQ was manually annotated by three annotators: Annotator A, a requirements engineer; Annotator B, an expert in NLP; and Annotator C, the lead author of this paper who is a PhD candidate investigating the use of NLP techniques in RE. Working independently, the annotators carried out the labelling task over a period of six weeks.

In order to assess the consistency of annotations between our annotators, inter-annotator agreement (IAA) was calculated based on the F-score, i.e., the harmonic mean of precision (P) and recall (R). In this case, F-score was computed for each of Annotators A and B, treating Annotator C's annotations as gold standard. Here, true positives are annotations from each of Annotator A and B that overlap with those of Annotator C, while false negatives are those which were missed. The false positives, meanwhile, are comprised of the annotations from each of Annotator A and B which are not in Annotator C's annotations. We obtained an F-score of 83% for Annotator A and 72% for Annotator B. These results indicate that there is a more than satisfactory level of consistency between our annotators, implying that the semantically related requirements that they annotated, can be considered as reliable.

In producing the final set of annotations, we harmonised the labels provided by our three annotators to resolve any disagreements between them. That is, Annotator C revisited every discrepancy, e.g., a requirement pair for which Annotator A has a different opinion from Annotator B, or Annotator C assigned a label that differs from that of Annotators A and B. In cases where she was convinced that Annotator A's (or B's) perspective is more correct, she modified the labels assigned to requirement pairs. Annotations resulting from this harmonisation procedure formed the basis of the final SEM-REQ data set.

4.2 Baseline System

To evaluate the proposed F2F method further we implemented a baseline system underpinned by pre-trained word embeddings, i.e., Google's Word2Vec model[4] which was trained on approximately 100 billion words from a corpus of Google News articles.

Given a pair of requirement statements, the text of each statement was firstly preprocessed through tokenisation, stemming and stop-word removal. The pre-trained word embedding vector for each token in a statement is then retrieved, resulting in n vectors (corresponding to n unique tokens in a statement). We note that we simply ignored out-of-vocabulary words, i.e., tokens for which the word embedding vectors do not exist in the

[4]https://code.google.com/archive/p/word2vec/

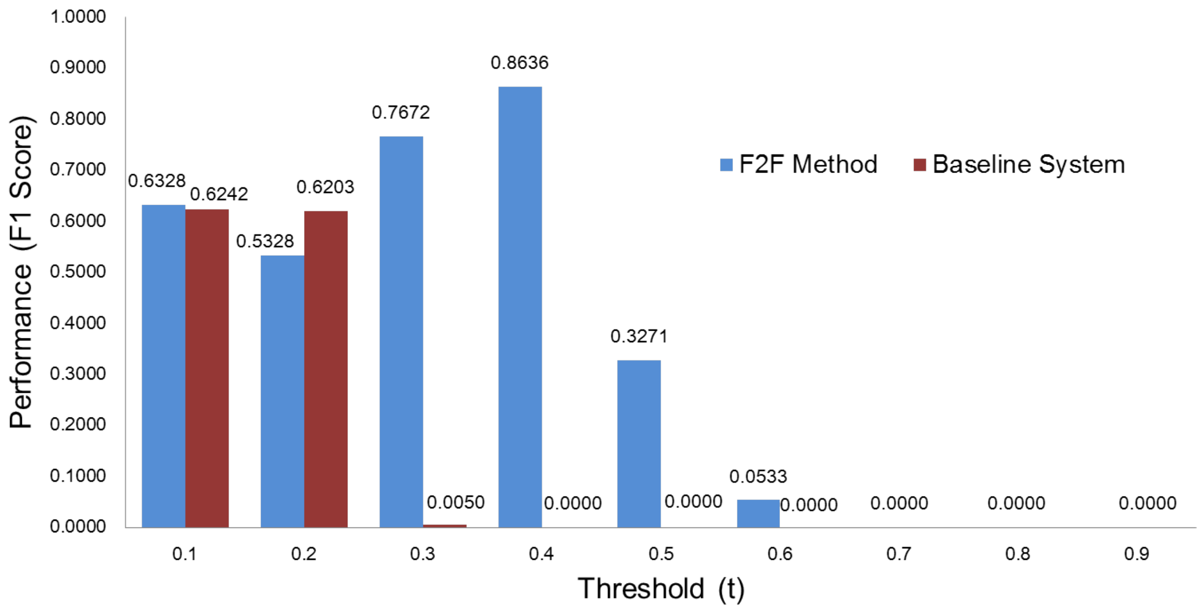


Figure 3: Performance based on F-scores obtained by the baseline system (blue-coloured bars) and F2F method (green-coloured bars) relative to SEM-REQ, depending on the minimum cosine similarity score (on the x-axis) that determines if statements in a pair are considered as semantically related or not.

Word2Vec model. The vectors retrieved for each requirement statement are then averaged, resulting in a vector representation of the statement.

Finally, the semantic relatedness between a pair of requirement statements was determined by computing the cosine similarity between their vector representations. As with our proposed method, the above-mentioned steps were implemented using the Gensim, NLTK, and numpy Python packages.

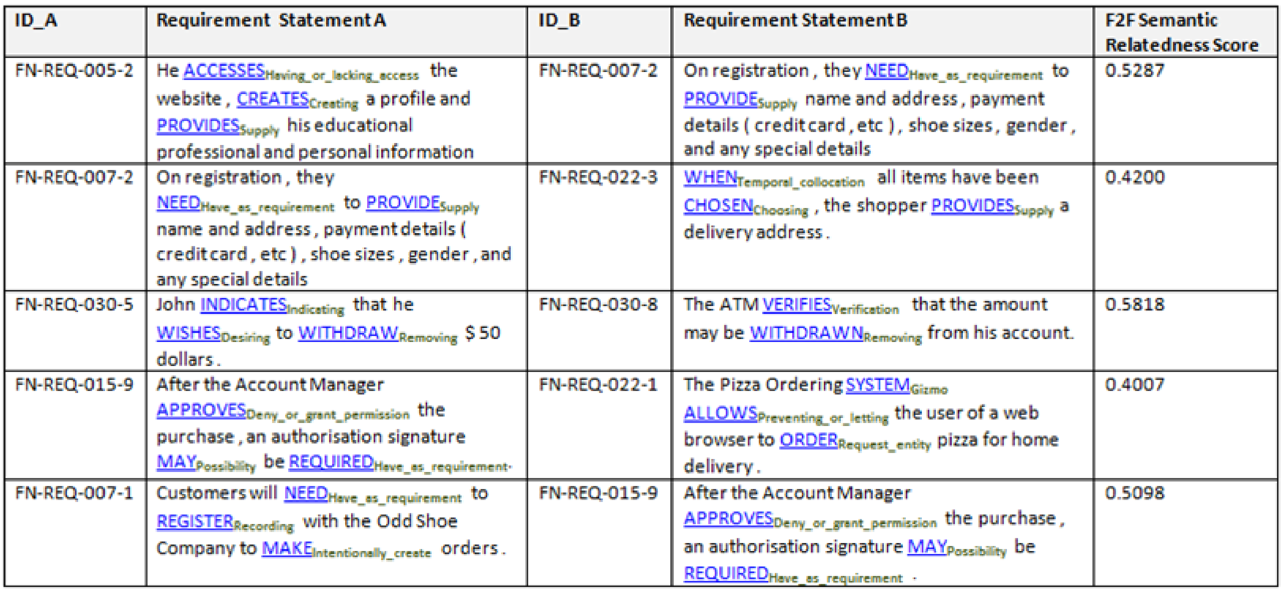
4.3 Results

We compared the F2F method with the baseline system by applying each of them to the same 1,770 pairs of requirement statements extracted from the SEM-REQ data set. The result obtained, a score s from each of the approaches, corresponds to the average of cosine similarities over the represented embeddings between requirement statements in a pair, where $0 \leq s \leq 1$. To help us determine what values of s can be considered as indicative of semantic relatedness, we experimented with different values of a minimum score or threshold t, whereby we say that the two statements in a pair are semantically related if $s \geq t$, and not semantically related otherwise.

For each value of t that was explored, ranging from 0.1 to 0.9 in increments of 0.1, a round of performance comparison was conducted based on the F-scores, obtained by each of the F2F method and the baseline system against the manual labels in SEM-REQ. The results of this comparison are shown in Figure 3.
It can be observed that while the F2F method obtained an F-score as high as 86.36% (with $t = 0.4$), the best performance obtained by the baseline system is only 62.42% (with $t = 0.1$). It is also worth noting that the baseline system is unable to detect semantically related statements unless only very low cosine similarity scores, i.e., those between 0.1 and 0.3, are taken into account. This implies that the cosine similarity scores obtained using the baseline system contradict the intuition that scores closer to 0 indicate lower semantic relatedness.

Examples of results obtained by the F2F method are shown in Figure 4. Although the statements in any of the five pairs shown are lexically different, our proposed F2F method detected them as semantically related, consistent with the judgement of human experts on the same pairs. In the first pair, for example, both statements explain a process for registering a new user (or customer) with different software systems. However, personal details (e.g., name and address of the user) required by the registration process were specified in Statement B, which is not the case in Statement A.

By detecting semantically related requirement statements, the F2F method can help in tracing connected events that a system is involved in. The statements of the third pair in Figure 4, for instance, discuss two events involving a cash withdrawal machine (the user specifying his desired amount, and the machine confirming that



ID_A	Requirement Statement A	ID_B	Requirement Statement B	F2F Semantic Relatedness Score
FN-REQ-005-2	He ACCESSES$_{\text{Having\_or\_lacking\_access}}$ the website , CREATES$_{\text{Creating}}$ a profile and PROVIDES$_{\text{Supply}}$ his educational professional and personal information	FN-REQ-007-2	On registration , they NEED$_{\text{Have\_as\_requirement}}$ to PROVIDE$_{\text{Supply}}$ name and address , payment details (credit card , etc) , shoe sizes , gender , and any special details	0.5287
FN-REQ-007-2	On registration , they NEED$_{\text{Have\_as\_requirement}}$ to PROVIDE$_{\text{Supply}}$ name and address , payment details (credit card , etc) , shoe sizes , gender , and any special details	FN-REQ-022-3	WHEN$_{\text{Temporal\_collocation}}$ all items have been CHOSEN$_{\text{Choosing}}$, the shopper PROVIDES$_{\text{Supply}}$ a delivery address .	0.4200
FN-REQ-030-5	John INDICATES$_{\text{Indicating}}$ that he WISHES$_{\text{Desiring}}$ to WITHDRAW$_{\text{Removing}}$ \$ 50 dollars .	FN-REQ-030-8	The ATM VERIFIES$_{\text{Verification}}$ that the amount may be WITHDRAWN$_{\text{Removing}}$ from his account.	0.5818
FN-REQ-015-9	After the Account Manager APPROVES$_{\text{Deny\_or\_grant\_permission}}$ the purchase , an authorisation signature MAY$_{\text{Possibility}}$ be REQUIRED$_{\text{Have\_as\_requirement}}$.	FN-REQ-022-1	The Pizza Ordering SYSTEM$_{\text{Gizmo}}$ ALLOWS$_{\text{Preventing\_or\_letting}}$ the user of a web browser to ORDER$_{\text{Request\_entity}}$ pizza for home delivery .	0.4007
FN-REQ-007-1	Customers will NEED$_{\text{Have\_as\_requirement}}$ to REGISTER$_{\text{Recording}}$ with the Odd Shoe Company to MAKE$_{\text{Intentionally\_create}}$ orders .	FN-REQ-015-9	After the Account Manager APPROVES$_{\text{Deny\_or\_grant\_permission}}$ the purchase , an authorisation signature MAY$_{\text{Possibility}}$ be REQUIRED$_{\text{Have\_as\_requirement}}$.	0.5098

Figure 4: Examples of results obtained by the proposed method on selected requirement statements in the SEM-REQ data set. The titles of semantic frames evoked by the statements are shown as subscripts appearing after the corresponding lexical units.

said amount can be withdrawn), with the latter having a dependency on the former. Tracing such relations within a requirement document is desirable, as it can aid, for example, in identifying dependency links between system features.

The obtained relatedness scores as presented in Figure 4 indicate various grades of semantic relatedness. For example, in the third requirement pair (which has the highest relatedness score of 58.18% among the examples shown), the statements refer to similar (e.g., *Removing* in the withdrawal sense) and connected events, e.g., *Verification* and *Indicating* (the user's bank account), where the frames share LUs. Meanwhile, in the fourth requirement pair (which has the lowest relatedness score of 40.07% among the examples), the frames in the two statements are not similar although they are semantically connected (i.e., *Deny_or_grant_permission* and *Preventing_or_letting*). Since these frames share contextual information in the trained frame embeddings, the statements in the requirement pair were correctly detected as semantically related. This indicates that, as desired, our proposed method accounts for contextual similarity in detecting semantic relatedness between requirement statements.

As shown in previous work, semantic frames are a promising means for capturing the meaning of requirements (e.g., [JM17]). Our encouraging results demonstrate that with careful selection of a threshold or minimum similarity score (for determining semantic relatedness), our proposed F2F method—which combines the strengths of semantic frames and embedding-based representations—can be applied to RE tasks such as software requirements analysis and traceability.

5 Conclusion

We presented a novel method for measuring semantic relatedness between software requirements, with a view to enhancing their traceability in requirements documents. The proposed method is based on the development of an embedding-based representation of semantic frames in FrameNet (i.e., frame embeddings), trained on a large corpus of user requirements, consisting of more than three million mobile application reviews. The performance of the method is encouraging as indicated by very good results (i.e., an F-score of 86.36%), outperforming a baseline system by 24 percentage points. In our immediate future work, we shall integrate this method with an RE tool for analysing and tracing semantic relatedness of software requirements. This in return, will aid in organising and grouping related system features described in requirements documents.

References

[ABNZ18a] Waad Alhoshan, Riza Batista-Navarro, and Liping Zhao. A framenet-based approach for annotating software requirements. In Tiago Timponi Torrent, Lars Borin, and Collin F. Baker, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, 2018. European Language Resources Association (ELRA).

[ABNZ18b] Waad Alhoshan, Riza Batista-Navarro, and Liping Zhao. Towards a corpus of requirements documents enriched with semantic frame annotations. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pages 428–431, 2018.

[AZBN18] Waad Alhoshan, Liping Zhao, and Riza Batista-Navarro. Using semantic frames to identify related textual requirements: An initial validation. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '18, pages 58:1–58:2, New York, NY, USA, 2018. ACM.

[Bak17] Collin F Baker. Framenet: Frame semantic annotation in practice. In *Handbook of Linguistic Annotation*, pages 771–811. Springer, 2017.

[BDVJ03] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003.

[BFL98] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

[DFFP18] Fabiano Dalpiaz, Alessio Ferrari, Xavier Franch, and Cristina Palomares. Natural language processing for requirements engineering: The best is yet to come. *IEEE Software*, 35(5):115–119, 2018.

[ECS18] Vasiliki Efstathiou, Christos Chatzilenas, and Diomidis Spinellis. Word embeddings for the software engineering domain. In *Proceedings of the 15th International Conference on Mining Software Repositories*, pages 38–41. ACM, 2018.

[FB01] Charles J Fillmore and Collin F Baker. Frame semantics for text understanding. In *Proceedings of WordNet and Other Lexical Resources Workshop, NAACL*, 2001.

[FDE$^+$17] Alessio Ferrari, Felice DellOrletta, Andrea Esuli, Vincenzo Gervasi, and Stefania Gnesi. Natural language requirements processing: a 4d vision. *IEEE Software*, (6):28–35, 2017.

[Fil76] Charles J Fillmore. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*, 280(1):20–32, 1976.

[GF94] Orlena CZ Gotel and CW Finkelstein. An analysis of the requirements traceability problem. In *Proceedings of IEEE International Conference on Requirements Engineering*, pages 94–101, 1994.

[Har54] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[HJD17] Elizabeth Hull, Ken Jackson, and Jeremy Dick. *Requirmenets Engineering.* Springer, London, 2017.

[JM17] Nishant Jha and Anas Mahmoud. Mining user requirements from application store reviews using frame semantics. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 273–287. Springer, 2017.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[MN15] Anas Mahmoud and Nan Niu. On the role of semantics in automated requirements tracing. *Requir. Eng.*, 20(3):281–300, 2015.

[MNX12] Anas Mahmoud, Nan Niu, and Songhua Xu. A semantic relatedness approach for traceability link recovery. In *2012 20th IEEE International Conference on Program Comprehension (ICPC)*, pages 183–192, 2012.

[SP18] Jennifer Sikos and Sebastian Padó. Using embeddings to compare framenet frames across languages. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 91–101, 2018.