JURNAL INFOTEL
Informatics - Telecommunication - Electronics
Website: http://ejournal.st3telkom.ac.id/index.php/infotel
ISSN: 2085-3688; e-ISSN: 2460-0997

Extracting software requirements-related information from online news using domText-WMDS

Mutia Rahmi Dewi[1,*], Indra Kharisma Raharjana[2], Daniel Siahaan[3], Nurul Jannah[4]
[1]Information Technology Major, Politeknik Negeri Padang
[2]Information System, Universitas Airlangga
[3]Department of Informatics, Institut Teknologi Sepuluh Nopember
[4]Electronics Engineering, International Islamic University Malaysia
[1]Limau Manis, Padang 25164, Indonesia
[2]Mulyorejo, Surabaya 60115, Indonesia
[3]Sukolilo, Surabaya 60111, Indonesia
[4]Gombak Street, Kuala Lumpur 53100, Malaysia
[*]Corresponding email: mutiarahmi@pnp.ac.id

Received 9 April 2023, Revised 6 June 2023, Accepted 26 June 2023

Abstract — Currently, few studies assess software requirements extraction from non-software artifacts. Most of the research in these related areas focuses on software artifacts such as project descriptions or user reviews as a source of requirements extraction. This research aims to identify relevant information to the software requirements from online news using the vector space model. This software requirements-related information can assist the analyst system in discovering the problem domain based on the lessons learned from stakeholders in online news. This research proposes DomText-WMDS to extract requirements-related information from online news. Using the domain specificity technique, we used the online news and public software requirements specification dataset to develop software-specific vocabulary. Then we expanded the specific vocabulary software to obtain more comprehensive results by building a vector space model from online news documents. This updated version of software-specific vocabulary could be used for basic filtering of software requirements-related information previously extracted using part-of-speech (POS) chunking. This study improved the performance for extracting software requirements-related information, with precision and recall at 61.09 % and 60.66 % compared to the domain specificity approach that only obtained 43.34 % and 40.78 %.

Keywords – domain specificity, online news, software requirements, specific vocabulary software, vector space model.

I. Introduction

In the process of requirements elicitation, analyst system usually use many resources to learn the domain knowledge of the software to be developed, such as interviews, observations, relevant documents, competitor analysis, legacy systems, and requirements reuse [1]–[3]. Each elicitation technique has advantages according to the characteristics of the software project [4]. Multiple elicitation techniques are typically performed simultaneously [2], [5]. Because of the uniqueness of software projects, the challenges that arose in the elicitation of software requirements are also diverse, such as limited access and engagement of the stakeholders, limited project time, and new and unique domain problems [5]–[11]. Often, analyst systems will look for alternative sources of elicitation to enrich domain knowledge to complete the software requirements.

One source of elicitation for software requirements that have yet to be explored is online news [12], which offers information about stakeholders' desires and lessons learned from a specific event [13]. Although it should be underlined that not all online news can be used as a source of elicitation, it is necessary to select topics to obtain relevant knowledge domains. The selection of relevant topics is also commonly applied to other sources in the requirements elicitation process, such as Twitter [14], [15], user app reviews [16], [17], and community discussions [18]. The use of online news as a source of needs is suitable for situations where it requires an understanding of a new problem domain, a problem in the public domain, and few related applications are available. For example, when we want to compile requirements for making related applications for handling natural disasters such as earthquakes and tsunamis, pandemics such as COVID-19, or dengue fever.

Obtaining domain knowledge from online news for the requirements elicitation process can help the analysis system acquire information related to domain knowledge quickly and minimize stakeholder involvement. Although stakeholder involvement is vital in software development, it has become a common problem that stakeholder commitment could be more robust [19]. Online news contains information about who, what, and why, which are elements that establish user stories [12], [20]. User stories are requirements artifacts commonly used in agile software development [21]. This user story format can capture stakeholder/user information, their needs, and their reasons for wanting it [22], [23]. In the user story, requirements are divided into four types: hard goal, soft goal, task, and capabilities [24]. Regarding content, online news is expected to be extracted as hard-goal and soft-goal. Task and capabilities may not be dominantly extracted because online news generally does not contain technical detail, such as conventional software artifacts.

Extracting requirements from online news documents, which are not related to software, is a challenge. Most studies extract requirements from software documents (such as software requirements specification [25], project description [26], or interview transcripts [27]) or documents directly related to the software (*e.g.*, user reviews [28], Twitter [15], or online discussion [18]). In the extraction process of documents related to software, researchers usually used the method to get phrases using part-of-speech (POS) tagging or n-gram and then select phrases based on the level of importance [22], [29]. This method cannot be applied directly to non-software-related documents like online news. It is necessary to take a particular approach to determine whether the phrases relate to software requirements. Currently, limited studies discuss software extraction requirements-related information in online news. In the previous study, we proposed a conceptual model for user story extraction from online news for requirements elicitation [12]. The domain specificity approach has been tried to find software requirements-related information from online news with a precision of 40.78 % and recall of 49.94 % [13]. The current results are still much room for improvement to find and enrich the information related to software functionality.

This paper addresses this problem by improving the extraction of software requirements-related information from online news by implementing a vector space model. Adapting Chen *et al.* [30] using vector space to create a software engineering thesaurus, this study uses a similar approach to identify software requirements-related information from online news. We use two data collections, namely online news and functional requirements, to compare and obtain the domain specificity of a topic in online news and then expand it using a vector space model. The FastText model converts words in online news sentences into vector form. These word vectors help in finding semantically adjacent sentences to functional software. Software requirements-related information will be obtained from phrase extraction using POS chunking with the pattern "verb + noun phrases." We chose this pattern because it fits what forms in the user story format.

II. Research Method

The overview of the proposed method can be found in Fig. 1. The inputs are functional requirements statements from the SRS repository as a domain corpus and online news documents as a general corpus. The output of our approach is a list of software-related terms from online news related to software requirements. Our approach is called DomText-WMDS, which stands for Domain Specificity, FastText, and Word Mover's Distance Similarity. Our method primarily includes four main steps: 1) performing preprocessing and phrase detection, 2) building software vocabulary, 3) building vector space, and 4) extracting requirements-related information.

A. Dataset

This study used a collection of functional requirements documents and online news with the specific topic as a dataset. Functional requirements documents represented the software domain. In contrast, online news represented a general domain unrelated to software artifacts.

We obtained a collection of functional requirements from PURE, a dataset of public requirements [31]. The PURE dataset was a collection of software requirements specification (SRS) documents gathered from various online sources and had a PDF or doc file format. We only used functional requirements data; we marked off the data contained in the document to get functional requirements statements. The number of SRS documents contained in the dataset was 78. The average of functional requirements statements in the SRS document was 75. The number of functional requirements statements was 5,961, and the average sentence's length of functional requirements statements was 50.

Online news collection crawled from several news agencies: BBC, CNA, CNN, FOX News, Huffing Post, Medical News, National Geographic, NYTimes, Stuff, The Guardian, Times of India, USA Today, WHO, and Yahoo News. In this study, we selected the 'dengue epidemic' topic as a target domain from online news. The total number of sentences in online news documents included in this dataset was 37,446.

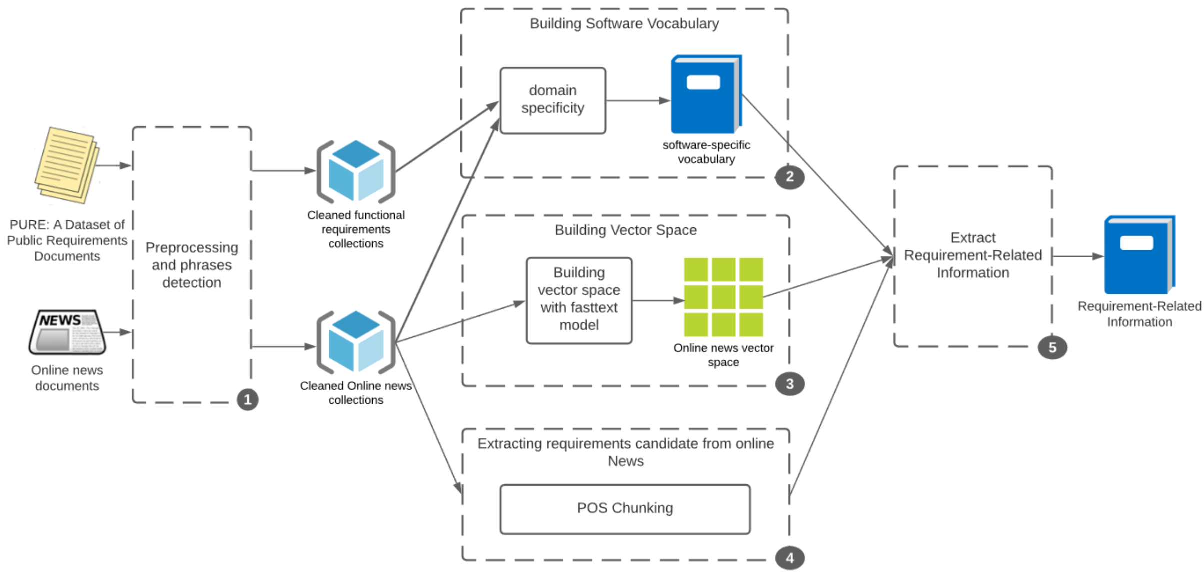

Fig. 1: Proposed method.

B. Preprocessing

1) Text cleaning

We performed text cleaning steps commonly used for preprocessing text. We applied it to functional requirements and online news collection. We first removed punctuation, number, and symbol from both datasets. Later, we carried out tokenization, stop words removal, and part-of-speech (POS) tagging.

2) The phrase detection

The phrase detection was formed iteratively based on the number of unigrams and bigrams using (1). Where w_i and w_{i+1} were two consecutive words, $count(w_i, w_{i+1})$ was the phrase frequency w_i, w_{i+1} in each document, and N was the total number of words in each document. δ was the discounting coefficient to eliminate phrases consisting of two words that were rarely formed. Two consecutive words were not from bigram phrases if they appeared as phrases with fewer phrases in each document than the specified discount coefficient. In this study, the value of the discount coefficient ten was determined based on previous research [32]. This approach aimed to recognize multi-word phrases in the text [32]. After the bigram phrases were formed, iterations were performed to detect the trigram and fourgram phrases.

$$score(w_i, w_{i+1}) = \frac{(count(w_i, w_{i+1}) - \delta) \times N}{count(w_i) \times count(w_{i+1})} \quad (1)$$

C. Building Software Specific Vocabulary

Identifying software vocabulary from a specific domain was accomplished using domain specificity [33]. This was achieved by comparing the token frequency from collections of the functional requirements and online news.

Eq. (2) was used to measure the domain specificity in terms. Where d and g respectively represented the specific and general corpus domains, $Pd(t)$ and $Pg(t)$ were the probabilities of token t from each corpus. For example, the probability of $Pd(t)$ of a token in a software-specific corpus d was calculated by dividing the token frequency $(c_d(t))$ by the total number of tokens (N_d) in the corpus. The specific token was the token that often appeared in a particular corpus of software but rarely appeared in the general corpus.

$$\text{domain specificity} = \frac{Pd(t)}{Pg(t)} = \frac{\frac{c_d(t)}{N_d}}{\frac{c_g(t)}{N_g}} \quad (2)$$

Based on these tokens, we looked for n-grams from the collections of functional requirements and online news. So, this stage produced a software-specific vocabulary containing a list of n-grams from a combination of the general and special corpus.

D. Building Vector Space

In recent years, vectors have been used to present words to capture information such as analogies and semantics. This approach can discover hidden information that cannot be obtained from dictionary-based methods. We used vector space to obtain words with semantic closeness to software-specific vocabulary. Software-specific vocabulary would be used as input to expand it to be more wide-ranging. This new vocabulary was used for extracting software requirements-related information from online news.

The online news collection was converted into word vectors. The vector space was created using the FastText model [35]. This model included state-of-the-art algorithms for examining word embedding using a neural network model. The basic intuition of this

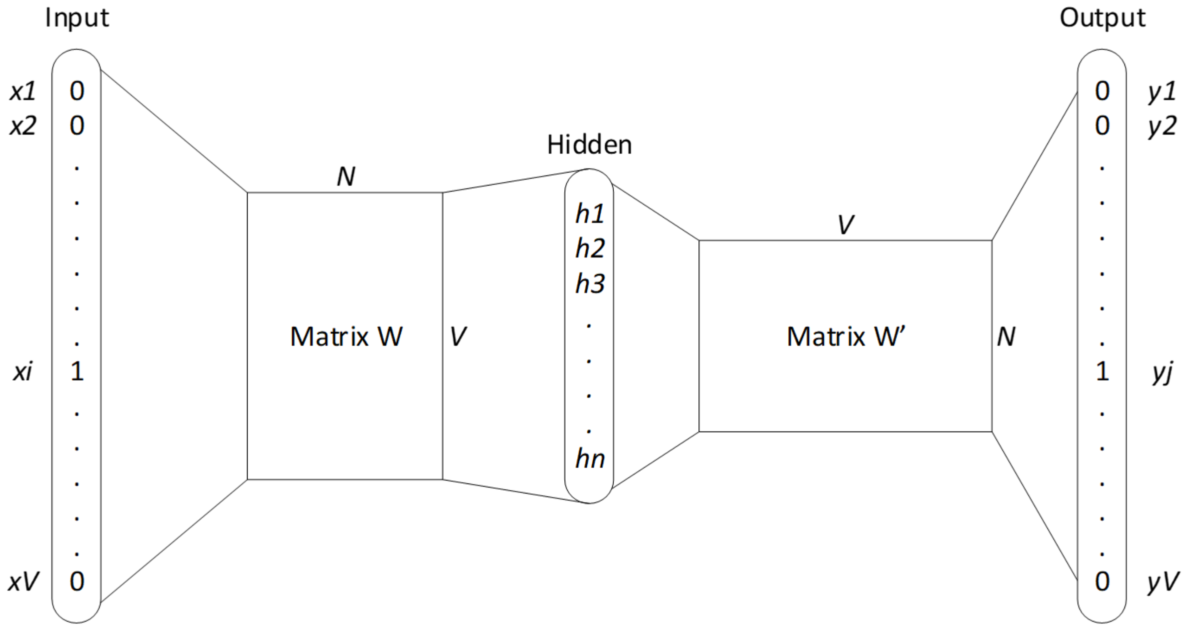

Fig. 2: One hot encoding architecture of fast text model [34].

algorithm was that words appearing in the same context had the same meaning. Therefore, the representation of each word could be defined in words that frequently co-occurred.

FastText model used a bag of character n-gram representations which acted as an input. Each word in the bag of character n-gram was enclosed in parentheses $<>$. For example, the online news corpus had a "dengue" token. If represented in the bag of character n-gram with n = 2, the result is: $< d, de, en, ng, gu, ue, e >$. The FastText model had the advantage of being able to represent words out of the vocabulary; this was because FastText exploited sub-word information [36].

In implementing word embedding with the FastText model, the learning process was carried out from the online news collection. It was necessary to determine the parameters in FastText first. The parameters of the FastText model were the dimensions and size ranges for the sub-words. Dimensions controlled the size of the vector. The larger the vector, the more information it could capture, which required more training data. Sub-words were all the substrings in the word between the minimum size (minn) and the maximum size (maxn). In this study using parameters, namely: dimension = 100, minn = 2, and maxn = 5.

The following were the learning steps in building vector space:

a) Initialize random weights on the W and W' matrices. Based on the one-hot encoding in Fig. 2, the W matrix represents the weight that connects the input and hidden layer with the matrix size $V \times N$. At the same time, the W' matrix represents the weight that connects the hidden layer and the output with the matrix size $N \times V$. The V value represents the number of vocab who was learned. In this study, the vocab used was $14,069$. Meanwhile, the value of N represents the dimension parameters that have been previously defined.
b) Initialize the learning data by determining the index of each vocab. Each vocab had a unique index with the matrix size $V \times 1$. After determining the vocab index, the learning data could be formed by determining the context and target words to predict each vocab.
c) Calculate each vocab's probability based on the one hot encoding architecture, as shown in Fig. 2. The target word was used as an input matrix in forming a vector space. At this stage, the probability was calculated by multiplying the input matrix with the W matrix to get the matrix value in the hidden layer. After that, the hidden layer matrix was multiplied by the W' matrix to get the output matrix value (a). The output matrix calculated probability, namely the softmax value using (3).

$$\text{Softmax} = \frac{e^{a_i}}{\sum_j e^{a_j}} \tag{3}$$

d) Backward propagation or change of the value of vector W and W' based on the objective function. After the softmax matrix value was obtained, the difference between the softmax and the output matrix was calculated. This difference value calculated the weight of the delta matrix W and W'. So that the value of the matrix W and W' could change. The new W value matrix represented the vector space model, which contained the vector value of each vocab.

The vector space had a vector value with a $1 \times N$ matrix for each vocab in the model. The word vector representation in the vector space generated by the FastText model was used to extract semantically adjacent terms with software-specific vocabulary.

E. Extracting Requirements Candidate from Online News

Then, we used POS chunking to identify phrases that signified goals, tasks, or capabilities. These elements represented aspects of what was in user stories [37]. We used a pattern representing verb and noun phrases; this POS tag pattern was usually used to identify software features or requirements from software artifacts, such as user reviews from the Google app store [17]. The pattern of regular expression and POS tags used in this study are shown in Fig. 3; a detailed description of POS tags syntax can be checked at The Penn Treebank [38].

```
NC: {(<DT>+<JJ|JJS|JJR>*)*(<IN>*<NN|NNP|NNS|NNPS>)+}
VERB: {<VB|VBG|VBZ|VBD|VBN|VBP>+<IN>*}
WHAT: {<VERB|JJ|JJS|JJR>+<DT>*<NC|(RB|RBR|RBS)>}
```

Fig. 3: POS chunking pattern (NC: Noun Phrases, VERB: verb phrase, WHAT: aspect of what in user story—a candidate for software requirements-related information).

F. Extracting of Semantic Related Term

We extracted requirement-related information using Word Mover's Distance Similarity (WMDS). WMDS was a function for calculating the distance between two text documents and determining the semantic similarity value [39]. Two documents collection used in this study were terms in the software-specific vocabulary and terms in the general corpus. The implementation of this WMDS function used a vector model to capture the semantic similarity between words.

This method finds distance as the minimum cumulative weight between two documents. The cumulative minimum weight value is obtained by using (4). Where T_{ij} is denoted as how many words i to word j. Whereas $c(i, j)$ is the distance value between words vector (x_i dan x_j) from each document obtained from (5). Once the cumulative minimum weight value is obtained, the semantic likeness value between documents is obtained using (6).

$$\text{distance } = \min_{T \geq 0} \sum_{i,j=1}^{n} T_{ij} c(i, j) \tag{4}$$

$$c(i, j) = \|x_i - x_j\|^2 \tag{5}$$

$$WDMS = \frac{1}{1 + \text{ distance}} \tag{6}$$

An example of semantic-related term extraction is shown in Fig. 4. Before calculating the distance between two documents, model building was conducted on Word Mover's Distance Similarity method. The input used in this model building was the term in the online news collection and the word vector in the vector space. In addition, determining the threshold in the form of a minimum semantic score was issued in this method. In this case, we set a threshold of 0.8. The results of this training were the form of a model that was used as a function to produce candidate terms semantically related to the functional software. The results are shown in box number (1) of Fig. 4.

Semantically related terms were extracted to expand software-specific vocabulary and produce a list of terms relevant to the software's functionality in online news documents. Term extraction is completed by measuring the similarity of terms in the software-specific vocabulary (available in box number 2 of Fig. 4) with terms in the online news collection.

Each term t in the software-specific vocabulary was inputted into the Word Mover's Distance Similarity algorithm to discover the similarity of the term t to the term in the online news collection. The similarity value between sentences is found in 6. Distance and similarity calculations happen in box numbers 3 and 4 of Fig. 4.

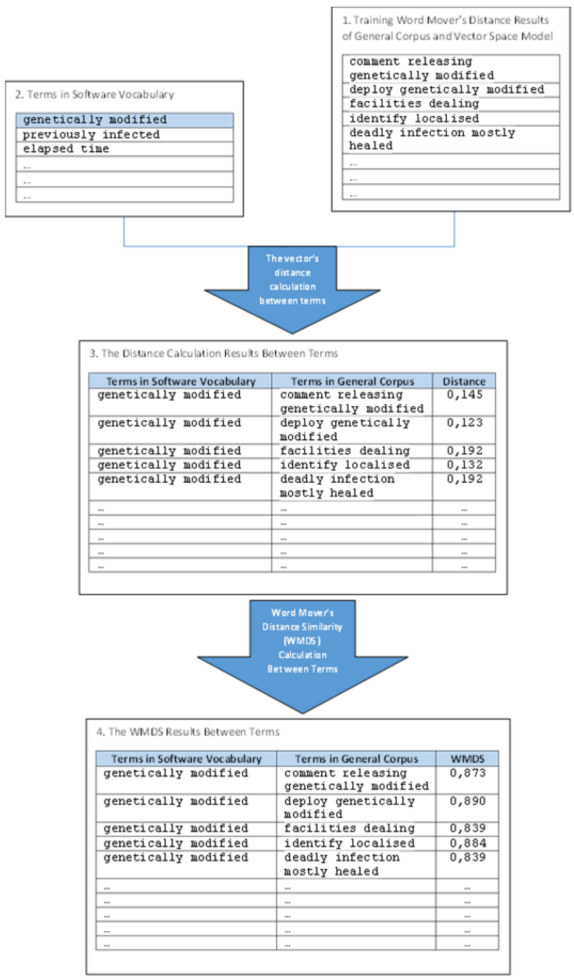

Fig. 4: Illustration of extracting semantic related software vocabulary from online news.

G. Illustration of extracting semantic related software vocabulary from online news

Requirement-related information is obtained from requirements candidates based on terms processed from DomText-WMDS. The requirements candidate has a verb and noun phrase format, according to the POS chunk pattern in Fig. 3. Meanwhile, the DomText-WMDS terms are n-gram (bigram, trigram, and fourgram phrases). Because we want a verb format followed by noun phrases as a form of requirement-related information, the format of terms obtained from DomText-WMDS needs to be modified. To get requirement-related information, we check to see if the requirements candidate contains terms from the DomText-WMDS; if they contain these terms, the requirements candidate are converted into requirement-related information.

H. Evaluation

The extraction results were used to test the reliability of the proposed method. The reliability of the proposed method was measured based on the accuracy of the extraction results with its ground truth. In this case, the ground truth was obtained from the answers/annotations of experts. The results of the proposed method were compared to the ground truth.

The experts in this evaluation were those qualified as analyst systems or software project developers. Experts could work as professionals or academics. Experts gave annotations in phrases, which were aspects of what the user story listed in online news documents. Before performing annotations, the experts were first explained the definition of software requirements-related information.

Since many news sentences need to be checked, a sample was randomly selected from the annotation set. Sampling was done by calculating the minimum amount of data to ensure that the estimated population was within the confidence interval at the confidence level [40]. The minimum amount of data was obtained through (7).

$$MIN = \frac{n_0}{1 + \frac{n_0 - 1}{\text{population size}}} \tag{7}$$

Where population size represented the number of populations, in this case, was the number of sentences in online news as a set of annotations. was obtained based on the selected confidence level and error margin values. The value of n_0 was calculated using (8).

$$n_0 = \left(Z^2 \times 0.25\right) / e^2 \tag{8}$$

Z is the confidence level z score, and e is the error margin. We used an error margin of 0.05 and a confidence level of 90 %. So we got 267 online news sentences that experts evaluated. The accuracy of the extraction results based on the degree of similarity with the answers/annotations of experts was measured using precision and recall.

III. RESULT

This section discusses the software-specific vocabulary, semantic related term, requirement-related information, and evaluation.

A. Software-Specific Vocabulary

The software-specific vocabulary was achieved by calculating domain specificity. There are 132 terms in the software-specific vocabulary that has been obtained. Some of the results of software-specific vocabulary shown in Table 1.

Table 1: Sample of Software-specific Vocabulary Produced for Dengue Case Study

No.	Token	Term
1	‘graph’, ‘displayed’	‘displayed graph’
2	‘modified’	‘genetically modified’
3	‘audit’, ‘keeps’, ‘trail’	‘keeps audit trail’
4	‘functionality’	‘software functionality’
5	‘verify’, ‘shall’	‘shall verify’

B. Semantic Related Term

The expansion of terms of software-specific vocabulary was carried out to obtain the semantically related terms. We look for the adjacent terms in Word Mover’s Distance training model for each term in the software-specific vocabulary. Some of the terminal expansion results that are semantically related are shown in Table 2. The similarity scores between software-specific vocabulary and terms extracted from online news were obtained from the WMDS function in (6). This score explained how closely the meanings of the two terms were semantically related.

C. Requirement-Related Information

Then we enriched the terms by looking for terms in the general corpus that were semantically adjacent to each term in the specific vocabulary software. Each term in software-specific vocabulary had several semantically adjacent terms. Of the $237,542$ n-grams in the general corpus containing the POS pattern Verb and Noun Phrase tags, $10,290$ semantically related terms were generated using FastText model. This term was used as a result of software-related information extraction. Some examples of online news sentences with the results of extracted related information software can be seen in Table 3.

D. Evaluation

The proposed method was evaluated by calculating the precision and recall values based on the extraction and ground truth results. Ground truth in each online news sentence was obtained from the results of recommendations by three experts. The evaluation results are shown in Table 4.

Table 2: The Expansion of Semantic Related Term from Software-specific Vocabulary

Software-specific Vocabulary	Term Extracted from Online News	Similarity Score	Semantic Related Term
'indicating start'	'creating situations'	0,83	'creating situations'
'message originating'		0,81	
'accept environmental'	'reintroduce vaccine'	0,81	'reintroduce vaccine'
'acknowledge receipt'	'seeking medical care'	0,80	'seeking medical care'
'used tires'	'determine suppression'	0,82	'determine suppression'
'system provides'		0,81	
'samples collected'	'implemented households places'	0,80	'implemented households places'
'checking rules'	'including purchase vehicles'	0,82	'including purchase vehicles'
'integrated vector'	'integrated vector management'	0,84	'integrated vector management'
'previously infected'	'implemented simultaneously effective control'	0,82	'implemented simultaneously effective control'
'porting original'	'planning improving sanitation'	0,80	'planning improving sanitation'
'message originating'	'investigating case'	0,83	'investigating case'

Table 3: Extraction Result from Online News

Online News's Sentences	Software-related Information
"We are creating situations that make things easier for those pests and pathogens that already cause us woe," she says.	'creating situations'
Despite the epidemic, the government has no immediate plans to reintroduce the vaccine to the public, Mark Timbal, spokesperson for the National Disaster Risk Reduction and Management Council told CNN Wednesday.	'immediate plans reintroduce vaccine', 'immediate plans', 'reintroduce vaccine'
A delay in seeking medical care in severe dengue cases is often related to deaths from dengue virus disease.	'seeking medical care'

Table 4: Evaluation Result

Method	Precision	Recall	Accuracy	F-Measure
POS chunking	29.85%	24.02%	46.57%	26.62%
Domain Specificity [13]	43.34%	40.78%	49.94%	42.02%
DomText-WMDS	61.09%	60.66%	65.18%	60.88%

The evaluation was done by comparing performance indicators for phrase extraction approaches in online news related to software (precision, recall, accuracy, and f-measure). The comparisons were POS chunking, domain specificity, and DomText-WMDS. POS chunking was commonly used to extract features from software-related artifacts such as user reviews [17], [41]. Meanwhile, domain specificity and DomText-WMDS were intended to obtain information related to software from non-related software artifacts such as online news.

The evaluation showed that DomText-WMDS could improve the precision and recall of extracted terms related to software requirements information. Compared with a previous study, the precision and recall could be improved from 43.34 % and 40.78 % to 61.09 % and 60.66 %. The precision and recall of DomText-WMDS outperformed the results of POS chunking which only produced values of 29.85 % and 24.02 %.

IV. Discussion

In a previous study [13], we tried to apply the domain specificity method to improve the precision and recall from the traditional approach (POS chunking). The results of the previous study indicated that domain specificity could increase precision and recall values. This increase suggested that a domain-specific approach could better obtain software-related information. In this study, we wanted to improve the precision and recall by expanding the terms obtained from the specificity domain using the vector space model. Vector space models have been widely used to obtain semantically related terms [30], [42]. The vector space model was built from online news collections.

In this study, the data used lists functional requirements in the Software Requirements Specification (SRS) document repository and online news documents. With phrase detection, domain specificity for software-specific vocabulary formation, vector space using the FastText model, and Word Mover's Distance Similarity algorithm for term similarity calculations were expected to produce a list of related information correlated to the software functionality.

The finding of this study was that the application of DomText-WMDS could improve the performance of precision and recall obtaining information relevant to software from online news. DomText-WMDS was an approach to obtaining information relevant to software from non-software-related artifacts by determining specific vocabularies with domain specificity. Then we expanded the vocabulary with a vector space model. In this study, we used FastText and WMDS to implement this.

The DomText-WMDS approach produced better precision and recall than using only domain specificity. In addition, the DomText-WMDS approach and domain specificity outperform the general approaches used in available software artifacts, such as POS chunking. This result was reasonable because the general approaches only looked at the POS pattern of the sentence without considering the domain context. This method has proven successful when used in artifacts related to software, such as user reviews, app descriptions, and Twitter [14], [21], [43]. However, when implemented in artifacts that were not related to software, the resulting outcomes are less than expected.

The extraction results were relevant to the software's functionality based on the extraction of requirements-related information that had been done using DomText-WMDS. It was because, in the extracted term, there were words that contained software functionality. These words were obtained from a collection of statements of functional requirements in the SRS repository. For example, the requirements-related terms 'creating the situation,' 'reported thousand people,' and 'recorded dengue' contained the words' creating,' 'reported,' and 'recorded,' commonly found in the SRS documents. This reveals that the SRS document impacts producing relevant information to software functionality from online news. Using the DomText-WMDS approach, we can find information related to software functionality requirements from online news based on its lesson learned. Domain specificity helped filter the term consist in online news and SRS documents. FastText model produces a words vector, which was used to extract related semantically terms with WMDS algorithm.

V. CONCLUSION

This study proposes extracting software-related information from online news using DomText-WMDS, which is helpful for requirements elicitation. We increased the precision and recall values compared to approaches that only used domain specificity. The result of this study achieved 60.97 % precision and 61.03 % recall. Limited studies existed on acquiring software requirement information from non-software sources. Most studies focused on software artifacts and used conventional approaches like part-of-speech or n-gram patterns, which were ineffective for extracting requirements from non-software sources like online news. Additional effort was required to filter software-related information from non-software artifacts.

Extracting software requirement information from online news was suitable for the requirements elicitation stage, particularly when facing challenges in stakeholder engagement or limited time. It could be as early knowledge for the analyst system before discussing it with stakeholders. Future studies should explore innovative approaches to enhance precision and recall. Further research is needed to fully understand and utilize information extracted from online news for analyst systems.

REFERENCES

[1] C. Pacheco, I. García, and M. Reyes, "Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques," *IET Software*, vol. 12, no. 4, pp. 365–378, 2018.

[2] P. A. Laplante and M. Kassab, *Requirements Engineering for Software and Systems*. New York: Auerbach Publications, 4 ed., 6 2022.

[3] K. Curcio, T. Navarro, A. Malucelli, and S. Reinehr, "Requirements engineering: A systematic mapping study in agile software development," *Journal of Systems and Software*, vol. 139, pp. 32–50, 2018.

[4] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Computers in Human Behavior*, vol. 51, pp. 915–929, 2015. Computing for Human Learning, Behaviour and Collaboration in the Social and Mobile Networks Era.

[5] C. Palomares, X. Franch, C. Quer, P. Chatzipetrou, L. López, and T. Gorschek, "The state-of-practice in requirements elicitation: An extended interview study at 12 companies," *CoRR*, vol. abs/2102.11556, 2021.

[6] D. A. P. Sari, A. Y. Putri, M. Hanggareni, A. Anjani, M. L. O. Siswondo, and I. K. Raharjana, "Crowdsourcing as a tool to elicit software requirements," *AIP Conference Proceedings*, vol. 2329, p. 050001, 02 2021.

[7] M. Bano, D. Zowghi, A. Ferrari, P. Spoletini, and B. Donati, "Teaching requirements elicitation interviews: an empirical study of learning from mistakes," *Requirements Engineering*, vol. 24, pp. 259–289, 5 2019.

[8] S. Bagheri, R. Kusters, J. Trienekens, and P. Grefen, "A reference model-based user requirements elicitation process: Toward operational business-it alignment in a co-creation value network," *Information and Software Technology*, vol. 111, pp. 72–85, 2019.

[9] L. M. Alves, G. Souza, P. Ribeiro, and R. J. Machado, "Longevity of risks in software development projects: a comparative analysis with an academic environment," *Procedia Computer Science*, vol. 181, pp. 827–834, 2021. CENTERIS 2020 - International Conference on ENTERprise Information Systems / ProjMAN 2020 - International Conference on Project MANagement / HCist 2020 - International Conference on Health and Social Care Information Systems and Technologies 2020, CENTERIS/ProjMAN/HCist 2020.

[10] W. Hussain, H. Perera, J. Whittle, A. Nurwidyantoro, R. Hoda, R. A. Shams, and G. Oliver, "Human values in software engineering: Contrasting case studies of practice," *IEEE Transactions on Software Engineering*, vol. 48, no. 5, pp. 1818–1833, 2022.

[11] H. Dar, M. I. Lali, H. Ashraf, M. Ramzan, T. Amjad, and B. Shahzad, "A systematic study on software requirements elicitation techniques and its challenges in mobile application development," *IEEE Access*, vol. 6, pp. 63859–63867, 2018.

[12] I. K. Raharjana, D. Siahaan, and C. Fatichah, "User story extraction from online news for software requirements elicitation: A conceptual model," in *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 342–347, 2019.

[13] M. Rahmi Dewi, I. Kharisma Raharjana, D. Siahaan, and C. Fatichah, "Software requirement-related information extraction from online news using domain specificity for requirements elicitation: How the system analyst can get software requirements without constrained by time and stakeholder availability," in *Proceedings of the 2021 10th International Conference on Software and Computer Applications*, ICSCA '21, (New York, NY, USA), p. 81–87, Association for Computing Machinery, 2021.

[14] M. Nayebi, H. Cho, and G. Ruhe, "App store mining is not enough for app improvement," *Empirical Software Engineering*, vol. 23, pp. 2764–2794, 2 2018.

[15] D. Martens and W. Maalej, "Extracting and analyzing context information in user-support conversations on twitter," in *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pp. 131–141, 2019.

[16] Y. Liu, L. Liu, H. Liu, and X. Yin, "App store mining for iterative domain analysis: Combine app descriptions with user reviews," *Software: Practice and Experience*, vol. 49, no. 6, pp. 1013–1040, 2019.

[17] I. K. Raharjana, V. Aprillya, B. Zaman, A. Justitia, and S. S. M. Fauzi, "Enhancing software feature extraction results using sentiment analysis to aid requirements reuse," *Computers*, vol. 10, no. 3, 2021.

[18] J. Ali Khan, L. Liu, and L. Wen, "Requirements knowledge acquisition from online user forums," *IET Software*, vol. 14, no. 3, pp. 242–253, 2020.

[19] M. Shameem, B. Chandra, R. R. Kumar, and C. Kumar, "A systematic literature review to identify human related challenges in globally distributed agile software development: towards a hypothetical model for scaling agile methodologies," in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pp. 1–7, 2018.

[20] Y. Wautelet, S. Heng, D. Hintea, M. Kolp, and S. Poelmans, "Bridging user story sets with the use case model," in *Advances in Conceptual Modeling* (S. Link and J. C. Trujillo, eds.), (Cham), pp. 127–138, Springer International Publishing, 2016.

[21] I. K. Raharjana, D. Siahaan, and C. Fatichah, "User stories and natural language processing: A systematic literature review," *IEEE Access*, vol. 9, pp. 53811–53826, 2021.

[22] G. Lucassen, M. Robeer, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "Extracting conceptual models from user stories with visual narrator," *Requirements Engineering*, vol. 22, pp. 339–358, 5 2017.

[23] I. K. Raharjana, F. Harris, and A. Justitia, "Tool for generating behavior-driven development test-cases," *Journal of Information Systems Engineering and Business Intelligence*, vol. 6, p. 27–36, Apr. 2020.

[24] Y. Wautelet, M. Velghe, S. Heng, S. Poelmans, and M. Kolp, "On modelers ability to build a visual diagram from a user story set: A goal-oriented approach," in *Requirements Engineering: Foundation for Software Quality* (E. Kamsties, J. Horkoff, and F. Dalpiaz, eds.), (Cham), pp. 209–226, Springer International Publishing, 2018.

[25] Y. Priyadi, A. Djunaidy, and D. Siahaan, "Requirements dependency graph modeling on software requirements specification using text analysis," in *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS)*, vol. 1, pp. 221–226, 2019.

[26] C. Li, L. Huang, J. Ge, B. Luo, and V. Ng, "Automatically classifying user requests in crowdsourcing requirements engineering," *Journal of Systems and Software*, vol. 138, pp. 108–123, 2018.

[27] P. Rodeghero, S. Jiang, A. Armaly, and C. McMillan, "Detecting user story information in developer-client conversations to generate extractive summaries," in *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pp. 49–59, 2017.

[28] Q. L. Sutino and D. O. Siahaan, "Feature extraction from app reviews in google play store by considering infrequent feature and app description," *Journal of Physics: Conference Series*, vol. 1230, p. 012007, jul 2019.

[29] S. Nasiri, Y. Rhazali, M. Lahmer, and N. Chenfour, "Towards a generation of class diagram from user stories in agile methods," *Procedia Computer Science*, vol. 170, pp. 831–837, 2020. The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

[30] X. Chen, C. Chen, D. Zhang, and Z. Xing, "Sethesaurus: Wordnet in software engineering," *IEEE Transactions on Software Engineering*, vol. 47, no. 9, pp. 1960–1979, 2021.

[31] A. Ferrari, G. O. Spagnolo, and S. Gnesi, "Pure: A dataset of public requirements documents," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 502–505, 2017.

[32] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013.

[33] Y. Park, S. Patwardhan, K. V. Ramanan, and S. C. Gates, "An empirical analysis of word error rate and keyword error rate," in *Interspeech*, 2008.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[35] J. Choi and S.-W. Lee, "Improving fasttext with inverse document frequency of subwords," *Pattern Recognition Letters*, vol. 133, pp. 165–172, 2020.

[36] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2017.

[37] Y. Wautelet, D. Gielis, S. Poelmans, and S. Heng, "Evaluating the impact of user stories quality on the ability to understand and structure requirements," in *The Practice of Enterprise Modeling* (J. Gordijn, W. Guédria, and H. A. Proper, eds.), (Cham), pp. 3–19, Springer International Publishing, 2019.

[38] J. Zhou and H. Zhao, "Head-Driven Phrase Structure Grammar parsing on Penn Treebank," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 2396–2408, Association for Computational Linguistics, July 2019.

[39] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 957–966, PMLR, 07–09 Jul 2015.

[40] R. Singh and N. S. Mangat, *Elements of Survey Sampling*. Springer Dordrecht, 1 ed., 3 2013.

[41] T. Johann, C. Stanik, A. M. Alizadeh B., and W. Maalej, "Safe: A simple approach for feature extraction from app descriptions and app reviews," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 21–30, 2017.

[42] R. Singh and S. Singh, "Text similarity measures in news articles by vector space model using nlp," *Journal of The Institution of Engineers (India): Series B*, vol. 102, pp. 329–338, 11 2021.

[43] J. Dabrowski, E. Letier, A. Perini, and A. Susi, "Finding and analyzing app reviews related to specific features: A research preview," in *Requirements Engineering: Foundation for Software Quality* (E. Knauss and M. Goedicke, eds.), (Cham), pp. 183–189, Springer International Publishing, 2019.