

Combining VSM and BTM to Improve Requirements Trace Links Generation

Bangchao Wang, Rong Peng, Zhuo Wang and Yaxin Zhao

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

September 29, 2019

Combining VSM and BTM to Improve Requirements Trace Links Generation

Bangchao Wang, Rong Peng*, Zhuo Wang, Yaxin Zhao
School of Computer Science, Wuhan University, Wuhan 430072, China
E-mail: {wangbc, rongpeng, wz2017, 2016301500098}@whu.edu.cn

Abstract—Trace links between software artifacts provide available traceability information and in-depth insights for different stakeholders. Unfortunately, establishing trace links is a fallible, tedious, and labor-intensive task. To alleviate these problems, many Information Retrieval (IR) methods, such as Vector Space Model (VSM), Latent Semantic Indexing (LSI) and their variants, have been proposed to establish trace links automatically. In recent years, short-text artifacts (or even lack of documentation) become a new trend as more and more software systems are developed abiding by agile methodologies. It makes the effects of traditional IR-based trace links generation methods even worse. In this paper, Biterm Topic Model (BTM), which is good at dealing with short text, is introduced to solve the problem. A hybrid method combining VSM and BTM is proposed to generate requirements trace links. The empirical experiments conducted on three real and frequently-used datasets indicate that the hybrid method can achieve better performance, and the results can reach the “acceptable level” directly.

Keywords—requirements traceability, information retrieval, vector space model, biterm topic model, short-text artifacts

I. INTRODUCTION

Requirements traceability (RT) is defined as “the ability to describe and follow the life of a requirement, in both a forward and backward direction [1]. In other words, it can provide visibility of the required aspects of the software and system development process, which contributes to a better understanding of the software system under development [2]. Thus, RT is one of the most important Requirements Engineering (RE) activities. Practically, traceability information has been proven vital to a wide variety software engineering activities, such as requirements consistency checking [3], change impact analysis [4], software reuse [5], and verification and validation (V&V) [6].

In early research and practice, RT is often accomplished by linking requirements to various software artifacts (e.g., design documents, source codes, and test cases) manually through a requirements traceability matrix [7]. However, as software systems evolve over time, RT activities are always time-consuming, tedious, and fallible [8]. As described in our previous work, determining how to improve the automatic degree and efficiency of a consistent and effective tracing process is an important challenge for both academia and industry [9].

To overcome this challenge, Information Retrieval (IR) techniques have been introduced and become the most popular techniques in the area of trace links generation [9]. A typical process of generating trace links by IR-based techniques for natural language artifacts, generally consists of the following steps: document preprocessing, candidate link generating, analyzing and refining [5]. After preprocessing, trace links can be automatically established using IR-based models, such as Vector Space Model (VSM)

[10, 11, 12], Latent Semantic Indexing (LSI) [13, 14, 15], and Probability Model (PM) [16, 17, 18].

In IR-based requirements tracing methods, requirements documents and target artifacts are usually regarded as queries and documents, respectively. And these methods aim to match a query of keywords with a set of objects in the software repository and rank the retrieved objects based on how relevant they are to the query using a predefined similarity measure [19, 20]. The tenet underlying IR-based tracing methods is that artifacts having a high textual similarity probably share several concepts, so they are likely good candidates to be traced from one another [20, 21, 22]. This tenet bases on the assumption that consistent terminologies have been used throughout the project’s lifecycle.

However, as projects evolve, new and inconsistent terminologies gradually emerge into the systems, which declines the performance of retrieval engines [20]. Besides, short-text, low quality text, and different expression preferences in different artifacts also negatively affect the performance of IR-based methods. To deal with all the problems, many strategies have been proposed to improve IR-based tracing methods [9, 23]. In an empirical study [24], the statistically analysis result shows several widely used IR-based tracing methods, which include VSM, LSI, Jensen-Shannon (JS), are almost equivalent. Meanwhile, another study [25] also draws an important conclusion that Latent Dirichlet Allocation (LDA) is able to capture some important information missed by other exploited IR methods, while its performance is lowest. These two conclusions indicate the opportunity to improve performance through combining different techniques, such as VSM and LDA, VSM and Relational Topic Model (RTM) [26]. However, two shortcomings still impede the combination techniques to get good performance on short-text artifacts tracing: firstly, they severely suffer from the severe data sparsity problem [27]; secondly, they cannot draw good results without enough learning corpus.

The empirical results introduced above motivates our work. A novel topic model which is good at dealing with short texts and lack of learning corpus problem.

In this paper, (1) Biterm Topic Model (BTM), good at dealing with short texts and lack of learning corpus, is introduced to generate trace links for the first time; and (2) a good-effect combination way—collection “union” operation (\cup)—is proposed to combine VSM and BTM to constitute candidate links set.

The remainder of this study is organized as follows. Section II provides background information, Section III reports the process of our approach, and Section IV presents the details of experiment. Experiment results are presented and discussed in Section V. Finally, in Section VI, the conclusions and future work are discussed.

*Corresponding author.

II. BACKGROUND

A. Starting Point for Discussion

How to evaluate the quality and usability of RT techniques is vital to propose new automatic RT techniques. The paper [30] proposed an acceptable and practical one according to their industrial practices, as shown in Table I. The standard also shows that candidate link lists with high recall and low precision are preferable to candidate link lists with high precision and low recall [30], as human analysts are much better in examining a given link list and determining whether it belongs to the answer set than they are in detecting whether the current set of links is sufficient.

TABLE I. STANDARDS FROM HAYES^[30]

Measure	Acceptable	Good	Excellent
Recall	60% — 69%	70% — 79%	80% — 100%
Precision	20% — 29%	30% — 49%	50% — 100%
Lag	3 — 4	2 — 3	0 — 2

The scalability of methods is another widely concerned issue. In other word, requirements tracing methods are expected to be able to achieve high quality for both “small” and “large” datasets. According to [30], a “small” dataset consists of 3000 combinatorial links or less. For example, a dataset consisting of 27 use cases and 87 test cases would have $27 \times 87 = 2349$ combinatorial links. Conversely, any dataset with more than 3000 combinatorial links is considered large.

In this work, we refer the quality evaluation standard of requirements tracing methods and the size boundary of dataset presented by [30], which is a starting point for discussion with researchers and practitioners.

B. Related works

In LDA, each document has a corresponding multinomial distribution over T topics and each topic has a corresponding multinomial distribution over the set of words in the vocabulary of the corpus. In [25], LDA has been introduced to generate trace links and capture some important information missed by other exploited IR methods, while its performance is low — the precision is less than 0.1. In [26], another topic model named RTM has been used to requirements traceability recovery. RTM is established with a foundation on LDA. Specifically, the process of modeling document-words distribution is identical to the LDA generative process. In their work, they propose a two-steps approach to combine similarity scores computed by two different IR methods for trace links generation. Firstly, the similarity scores of the two methods are mapped to a standard normal distribution. Secondly, the normalized scores are combined through a weighted sum. The value of confidence parameters for two IR methods need to be determined by users based on their experience. Since the homologous of their principle, two common shortcomings impede they generate trace links.

On the one hand, both LDA and RTM reveal the latent topics within the text corpus by implicitly capturing the document-level word co-occurrence patterns [28]. Since a lot of software artifacts are short texts, directly applying either LDA or RTM on this kind of artifacts will suffer from the severe data sparsity problem[29]. On the other hand, lacking

enough learning corpus may be another threat to impede the wide use of LDA and RTM. It is because insufficient learning corpus may not lead to good results in two different modeling processes.

Compared to these related works, our work introduce a new topic model named BTM to generate trace links for the first time. This model solve the severe data sparsity problem on short texts and the bad learning results problem on insufficient learning corpus. Besides, obtaining union (\cup) is proposed as a good-effect way to improve the candidate links set. The details of our approach will be presented in the next section.

III. OUR APPROACH

In this section, the overall process of our approach and the details of combining VSM and BTM to generate trace links will be presented.

A. Artifacts feature analysis

Generally, there are two kinds of text artifacts:

(1) Short text artifacts. This kind of artifacts only contain several words, such as “One sentence” text artifacts. For example, the requirement 103 in EBT only has one sentence: “A user shall register as a subscriber”.

(2) Long text artifacts. This kind of artifacts contains abundant contents to be traced on, such as structural text artifacts. For example, an artifact “use case” always includes use case name, summary, and description; and an artifact “test case” usually consists of test case name and pre/postconditions, as presented in TABLE II and III.

TABLE II. AN EXAMPLE OF USE CASE FROM EASYCLINIC

<i>UC01</i>	
Use case name:	input registry laboratory
Summary:	The Operator has been recognized by system and has all the data that characterize the the registry of the laboratory. The data in the S I O not be modified Success. The registry of the laboratory is properly inserted inside the S I O .
Description:	1.View the mask to enter information needed 2.Inserts data about the registry of laboratory 3.Confirm placement 4.Verify the data entered 5.Stores data 6.Notify operation it is finished with success

TABLE III. AN EXAMPLE OF TEST CASE FROM EBT

<i>TC141</i>	
Test case name:	Establish Trace (2.1.1)(2.2.1)
Preconditions& Postconditions:	Preconditions: Subscriber is registered Steps Subscriber establishes a trace between a UML artifact and a requirement. Postconditions: A trace is established between the UML artifact and the requirement.

As VSM is good at analyzing long text artifacts and BTM is good at analyzing short artifacts, different preprocessing processes are adopted according to their features.

B. BTM-based tracing method

As described in Section I, directly applying conventional topic models (e.g. LDA and RTM) on short texts do not work well as these models implicitly capture the document-level word co-occurrence patterns to reveal topics, and thus suffer from the severe data sparsity and lacking enough

learning corpus in short documents [28]. Thus, uncovering the topics within short-text artifacts and their relevance becomes a new challenge.

In [28], Yan propose a novel method for modeling topics in short texts, referred as biterm topic model (BTM). The key idea of BTM is to learn topics over short texts based on the aggregated biterms, namely word pairs, in the whole corpus to tackle the sparsity problem in single document. In other word, any two distinct words in an artifact are firstly extracted as a biterm. For example, in the short-text use case “The user can change password.”, if the stop words “The” and “can” are ignored after preprocessing, there are three biterms, i.e. “user change”, “user password”, and “change password”.

As presented in [28], suppose α and β are the Dirichlet priors. The specific generative process of the corpus in BTM can be described as follows:

1. For each topic z
 - (a) draw a topic-specific word distribution $\Phi_z \sim \text{Dir}(\beta)$
2. Draw a topic distribution $\theta \sim \text{Dir}(\alpha)$ for the whole collection
3. For each biterm b in the biterm set B
 - (a) draw a topic assignment $z \sim \text{Multi}(\theta)$
 - (b) draw two words: $w_i, w_j \sim \text{Multi}(\Phi_z)$

Figure 1 shows the biterm topic model, where θ is the topic probability distribution in the BTM corpus; Φ is the topic-word pair probability distribution; z is the serial number of topic for corresponding word pair, T is the number of topics; w_i, w_j are two words in the biterm; B is the number of word pairs in the entire corpus; all the word pairs shares the same topic distribution in corpus, each topic corresponds to a polynomial distribution of several word pairs and this Multinomial distribution is recorded as Φ , each word pair corresponds to a topic; α and β are the hyperparameters of Dirichlet prior distribution.

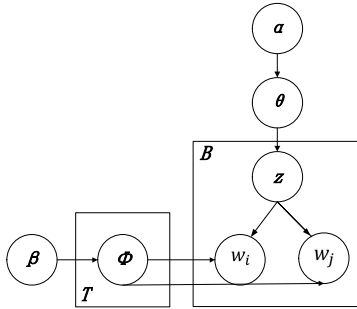


Fig. 1. Biterm topic model

As shown in Fig.2, the BTM-based trace links generation process can be divided into the following phases:

(1) Preprocessing. The preprocessing process is divided into two kinds: for short text artifacts, the pre-processing process contains stop words removal, part of speech tagging and word stemming; for long text artifacts, the upper pre-processing process is only executed on the most representative part of the artifact. For example, only the use/test case name is retained for the whole use/test case described in Table II/III.

(2) Biterms Extracting. In this step, any two distinct words got from the preprocessed source and target artifacts (namely the document collection) are firstly extracted as a biterm. And these biterms are used as the preprocessed corpus to train BTM.

(3) Biterm topic modeling. To use the preprocessed corpus training BTM, three parameters, including T , α and β , need to be set according to the experience related to the dataset. The outputs of this phase are: 1) topic distribution; 2) topic-word pairs distribution; and 3) document-topic distribution.

(4) Text similarity calculating. In this step, document-topic vectors must be established firstly. Based on these vectors, JS distance is used to represented the relevance between source and target artifacts. Note, it is different from cosine similarity since two texts are more similar when JS distance is smaller. After that, links are ranked according to JS distance.

(5) Candidate links generation. The generated links with top N similarity scores are selected as the candidate links.

Due to space limitations, we will not repeat the content about the specific probability calculation formulas of step (3) and JS distance calculation formulas of step (4). The details can be found in [28], where the BTM is proposed for the first time.

C. VSM-based tracing method

As shown in Fig.2, the VSM-based tracing method used in our approach can be divided into the following phases:

(1) Preprocessing. Both short artifacts and long artifacts are directly performed typical pre-processing steps: stop words removal, part of speech tagging and word stemming.

(2) Documents' vectors generation. A document d in the document collection is represented as a vector of keyword weights $d = (w_1, w_2, \dots, w_N)$, and the vocabulary of the entire collection is represented as (v_1, v_2, \dots, v_N) . The weight w_1 is calculated as the product of term frequency-inverse document frequency model (TF-IDF). Similarly, the query Q is also converted into a vector, represented as $q = (q_1, q_2, \dots, q_N)$.

(3) Text similarity calculation. And then, the relevance between document D and query Q is computed as the cosine of the angle between the vectors d and q , as represented in formula (1).

$$\text{sim}(d, q) = \cos(d, q) = \frac{\sum_{i=1}^N w_i \cdot q_i}{\sqrt{\sum_{i=1}^N w_i^2 \cdot \sum_{i=1}^N q_i^2}} \quad (1)$$

(4) Candidate links generation. After computing the cosine similarity, the retrieved objects are ranked based on the similarity scores. The artifacts with high textual similarity are likely good candidates to be traced from one another. In our approach, taking top N is adopt to determine the candidate links. In other words, the top N trace links with the highest scores will be selected as candidate links.

D. The hybrid method

Since complementarity exists between the VSM and BTM, “union” operation (\cup) has a great chance to improve the effect on combinations of precision and recall. The basic idea behind the hybrid method is that two IR methods, VSM and BTM, can be viewed as two experts who provide their expertise to generate candidate trace links. Both experts express judgments based on the textual similarity between two artifacts. The specific method is as following:

Suppose 1) the space size of a dataset is N ; 2) the dataset consisting of L true links and the count of the final candidate

links is nL ($n < N/L$); 2) the select ratios of VSM and BTM are λ and $1-\lambda$, respectively. Then, top $\lambda \times nL$ candidate links generated by VSM and top $(1-\lambda) \times nL$ candidate links generated by BTM are selected to construct the final trace links set. Note, after duplicate links removal, the count of the final trace links may be less than nL as some of the candidate links generated by VSM and BTM are same.

Finally, the candidate links generated in our hybrid method will be provided to analysts to obtain the final trace links as shown in Figure 2.

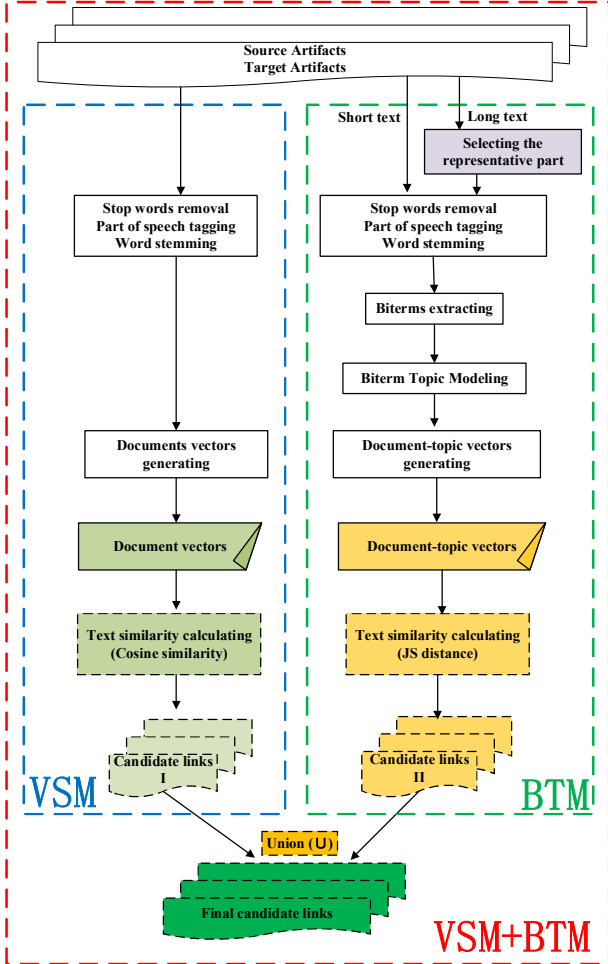


Fig. 2. The hybrid trace links generation method

IV. Experiment

This section describes the empirical experiments carried out on three real and frequently-used datasets to indicate that (1) it is reasonable to use different preprocess steps for VSM and BTM, (2) the hybrid method outperforms standard stand-alone IR methods, namely VSM and BTM.

It is note that the reason why LSI, JS and other similar methods are not regarded as comparison targets in our experiments is that VSM, LSI, JS and some similar methods are almost equivalent [24].

A. Research Question

To identify rationality and usefulness of the proposed hybrid method, the following two research questions need to be answered:

RQ1: Is it reasonable to use different preprocess steps for VSM and BTM?

RQ2: Does the hybrid method improve the quality of trace links generation and to what extent can this method achieve?

B. Datasets

Three datasets, WARC, EasyClinic, and EBT, are used to conduct the experiments. They are able to be downloaded at: <http://www.coest.org/>. The datasets are listed as following:

(1)WARC: This dataset includes 43 functional requirements (FRS), 21 non-functional requirements (NFR), 89 software requirements specification (SRS).

(2)EasyClinic: It is a small student-created dataset in English and Italian which contains diverse artifacts, including 30 Use Cases (UCs), 63 Test Cases (TCs), 20 Interaction Diagrams (IDs) and 47 Code Class descriptions (CCs). Note, only a subset of EasyClinic is chosen to conduct the experiment, as shown in Table III.

(3)EBT: It is an Event-Based Traceability (EBT) system, which contains 41 Requirements (Rqs), 25 Test Cases (TCs) and 52 Code Classes (CCs). Similarly, a subset of EBT is chosen to conduct the experiment, as shown in Table III.

Table IV shows the characteristics of each dataset used in the following experiments. Aiming to improve the availability of our method, the values of the parameters used in experiments are listed in Table V.

TABLE IV. DATASETS USED IN EXPERIMENTS

Dataset	Source	Target	Space (N)	True links (L)	Scale
WARC	43 FRS	89 SRS	3827	78	large
	21 NFR	89 SRS	1869	58	small
EasyClinic subset	30 UCs	63 TCs	1890	63	small
EBT subset	41 Rqs	25 TCs	1025	51	small

Note: “Space” represents the maximum counts of trace links.

TABLE V. PARAMETERS USED IN EXPERIMENTS

Dataset	T	α	β
WARC (FRS-SRS)	35	1.35	0.01
WARC (NFR-SRS)	35	1.35	0.01
EasyClinic (UC-TC)	20	1.25	0.01
EBT (Requirements-TC)	20	1.25	0.01

Note: $\lambda = 0.7$ is assigned to generate final trace links.

C. Quality Measures

Precision (P) and Recall (R) are the standard IR metrics to assess the quality of different requirements tracing techniques. Precision measures accuracy and recall measures coverage [17]. The combinations of precision and recall are used to evaluate the requirements tracing methods based on Hayes’ standard described in Section II.

V. RESULTS AND DISCUSSION

In the following section, the findings of the experiments are clarified. After that, the potential threats are discussed.

A. Results

Before discussing and analyzing the results for **RQ1** and **RQ2**, some marks shown in Figure 3, 4, 5, 6 and 7 are illustrated.

As shown in Table I, the “Acceptable”, “Good”, and “Excellent” results must have a recall rate more than 60% and a precision rate no less than 20%. Therefore, in Figure 3

to 7, three areas, distinguished by green, blue, and pink dotted box, in the upper right corner of the coordinate axis are used to represent these three types of results, which are superimposed as “effective areas”.

For **RQ1**, experiments have been just performed in EasyClinic and EBT, as the WARC has no long-text artifacts. As Figure 3 shows, the quality of VSM on “long-text” (the preprocessing step is following the first step described in Section 3C) is obviously better than the quality of VSM on short-text ((the preprocessing step is following the first step described in Section 3B) in effective areas. However, BTM is better at dealing with short-text, as shown in [28]. Thus, it is reasonable to use different preprocess steps for VSM and BTM.

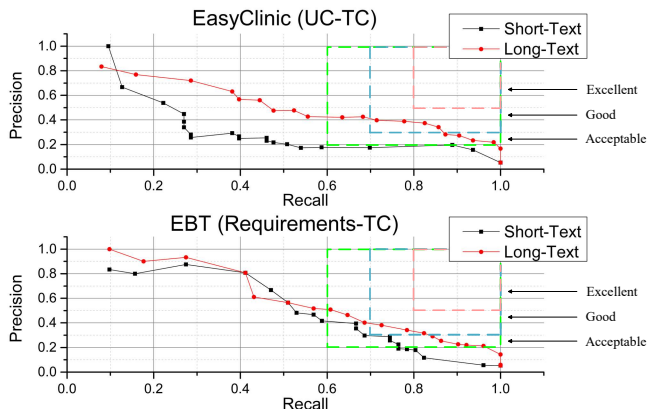


Fig. 3. The precision/recall distributions of VSM in short and long-text.

For **RQ2**, as shown in Figure 4, 5, and 6, all the red curves that fall in the effective area exceed the blue and black curves. In Figure 7, although both red and black curves can fall in the “Good” area, the quality of the red curve is better than the black one, which shows the combined method achieve the highest recall in the effective area.

Through the above analysis, the following conclusions can be summarized:

(1) The hybrid method outperforms VSM and BTM in all three datasets. In other words, the hybrid method improves the quality of trace links generation;

(2) The results of the hybrid method achieve “Good” level in EasyClinic, EBT and WARC (FRS-SRS) and “Acceptable” level in WARC (NFR-SRS).

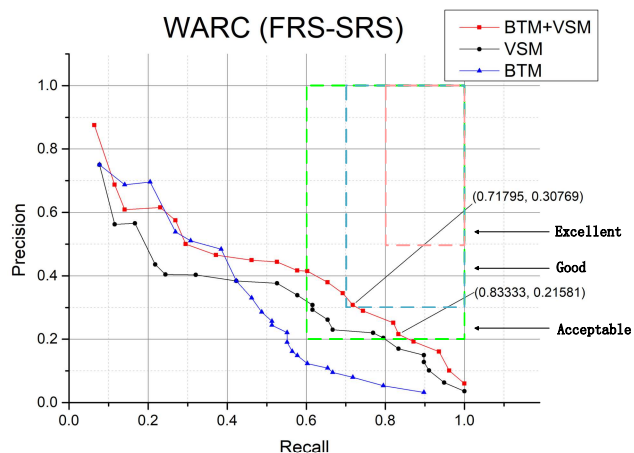


Fig. 4. The precision/recall curves of hybrid method and the standard VSM, BTM method in WARC (FRS-SRS).

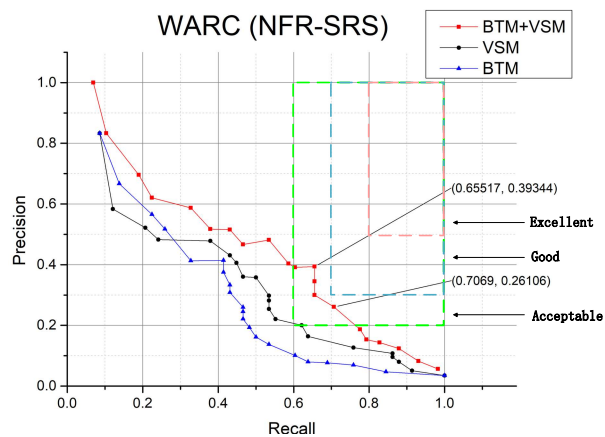


Fig. 5. The precision/recall curves of hybrid method and the standard VSM, BTM method in WARC (NFR-SRS).

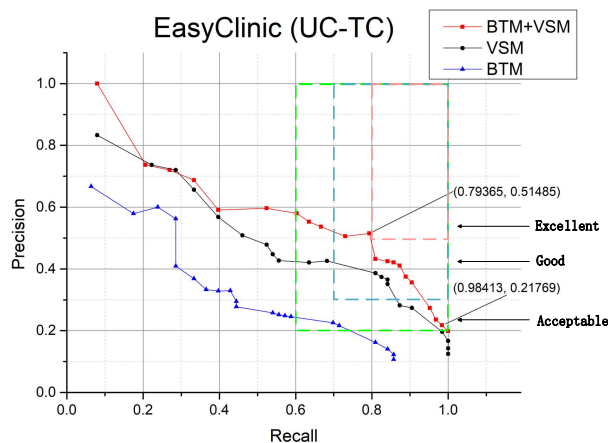


Fig. 6. The precision/recall curves of hybrid method and the standard VSM, BTM method in EasyClinic subset (UC-TC).

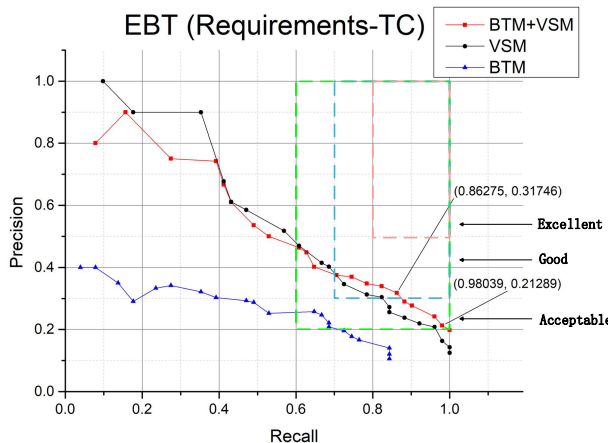


Fig. 7. The precision/recall curves of hybrid method and the standard VSM, BTM method in EBT subset (Requirements-TC).

B. Validity threats

This section aims to discuss the potential threats that influence the findings of this work. The following two aspects are discussed in this part.

Construct validity: (1) Since complementarity exists between the VSM and BTM, “union” operation (\cup) has a great chance to improve the effect. Therefore, it is reasonable to use these two methods to construct a hybrid tracing method. (2) As the result of RQ1 shows, VSM is good at analyzing long text artifacts and BTM is good at analyzing short artifacts, different preprocessing processes are adopted

according to their features. It ensures better effect can be obtained by the hybrid method. (3) The metrics used in our evaluation are recall and precision, which have been widely adopted for assessing the traceability accuracy of IR methods. Thus, we believe that they can sufficiently quantify the accuracy of three compared methods.

Conclusion validity: (1) In this work, experiment has been conducted on three real and frequently-used datasets. And, sufficient results and findings are summarized to illustrate the improvement. It illustrates the validity of our conclusions. (2) As the experiment results show, the hybrid method is able to achieve “Good” effect on small datasets as well as large datasets. The proposed hybrid method has a great chance to be generalized and introduced to other software systems and projects with various requirements and subsequent artifacts. (3) The validity of our experiment can also be affected by the chosen value of the parameter T , α , and β in the employed biterm topic model. We choose the value of three parameters based on our empirical evidence. In the future, we will obtain the optimal values by using some advanced techniques, such as optimization and machine learning.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, Biterm Topic Model (BTM), which is good at dealing with short texts and insufficient corpus, is introduced to generate trace links for the first time. After that, BTM is then combined with VSM to improve the effect of the standard IR methods. The experiments conducted on three real and frequently-used datasets in both large and small scale indicate that our method outperforms standard stand-alone IR methods, namely VSM and BTM.

In the future, we plan to perform detailed analysis about why BTM can complement some trace links for VSM-based tracing method. And then, some advanced parameter configuration techniques will be proposed to reduce the difficulty of using our method. Besides, all kinds of specific applicable scenarios for our method will be presented to improve the availability. Moreover, some refining strategies may be introduced to further improve the quality level.

ACKNOWLEDGMENT

This work is supported by the National Key Research and Development Plan of China under Grant No. 2017YFB0503702, 2016YFB0501801, National Natural Science Foundation of China under Grant No. 61170026.

REFERENCES

- [1] Gotel O, Finkelstein A (1994) An analysis of the requirements traceability problem. In: International conference on requirements engineering, pp 94–101
- [2] Gotel, O., Cleland-Huang, J., Hayes, J.H., Zisman, A., Egyed, A., Gru'nbacher, P., Dekhtyar, A., Antoniol, G., Maletic, J., M'ader, P., 2012. Traceability fundamentals. In: Software and Systems Traceability. Springer, pp. 3–22
- [3] Sultanov H , Hayes J H , Kong W K . Application of swarm techniques to requirements tracing[J]. Requirements Engineering, 2011, 16(3):209-226.
- [4] von Knethen A (2002) Automatic change support based on a trace model. In: International workshop on traceability in emerging forms of software engineering
- [5] Spanoudakis G, Zisman A (2004) Software traceability: a roadmap. Handb Softw Eng Knowl Eng 3:395–428
- [6] Huffman-Hayes J, Dekhtyar A, Sundaram S (2006) Advancing candidate link generation for requirements tracing: the study of methods. IEEE Trans Softw Eng 32(1):4–19
- [7] Ramesh B, Jarke M (2001) Towards reference models for requirements traceability. IEEE Trans Softw Eng 27(1):58–93
- [8] Dekhtyar A, Huffman-Hayes J, Antoniol G (2007) Benchmarks for traceability? In: International symposium on grand challenges in traceability
- [9] Bangchao Wang , Rong Peng , Yuanbang Li , Han Lai , Zhuo Wang, Requirements traceability technologies and technology transfer decision support: A systematic review, The Journal of Systems & Software,146C (2018) pp.59-79.
- [10] Niu, N. , & Mahmoud, A. . (2012). Enhancing candidate link generation for requirements tracing: The cluster hypothesis revisited. Requirements Engineering Conference (RE), 2012 20th IEEE International. IEEE.
- [11] Kong, W. K. , & Hayes, J. H. . (2011). Proximity-based traceability: An empirical validation using ranked retrieval and set-based measures. International Workshop on Empirical Requirements Engineering. IEEE.
- [12] Mahmoud, A. , & Niu, N. . (2013). Supporting requirements traceability through refactoring. Requirements Engineering Conference. IEEE.
- [13] Lormans, M. , & Van Deursen, A. . (2006). Can LSI help reconstructing requirements traceability in design and test?. Conference on Software Maintenance & Reengineering. IEEE.
- [14] Mcmillan, C. , Poshyvanyk, D. , & Revelle, M. . (2009). Combining textual and structural analysis of software artifacts for traceability link recovery. Workshop on Traceability in Emerging Forms of Software Engineering. IEEE.
- [15] Wang, X. , Lai, G. , & Liu, C. . (2009). Recovering Relationships between Documentation and Source Code based on the Characteristics of Software Engineering. Elsevier Science Publishers B. V.
- [16] Zou, X. , Settmi, R. , & Cleland-Huang, J. . (2010). Improving automated requirements trace retrieval: a study of term-based enhancement methods. Empirical Software Engineering, 15(2), 119–146.
- [17] Zou, X. , Settmi, R. , & Clelandhuang, J. . (2006). Phrasing in Dynamic Requirements Trace Retrieval. International Computer Software & Applications Conference. IEEE Computer Society.
- [18] Zou, X. , Settmi, R. , & Cleland-Huang, J. . (2008). Evaluating the Use of Project Glossaries in Automated Trace Retrieval. International Conference on Software Engineering Research & Practice. DBLP.
- [19] Grzywaczewski A, Iqbal R (2012) Task-specific information retrieval systems for software engineers. J Comput Syst Sci 78(4):1204–1218
- [20] Mahmoud, A., & Niu, N. . (2015). On the role of semantics in automated requirements tracing. Requirements Engineering, 20(3), 281-300.
- [21] Antoniol G, Caprile B, Potrich A, Tonella P (2000) Design-code traceability for object-oriented systems. Ann Softw Eng 9(1–4):35–58
- [22] De Lucia A, Fasano F, Oliveto R, Tortora G (2007) Recovering traceability links in software artifact management systems using information retrieval methods. ACM Trans Softw Eng Methodol 16(4):13–50
- [23] Hu Chenghai,Peng Rong,Wang Bangchao. A survey of requirement tracking method based on information retrieval. Computer Applications and Software, 2017(10):26-34.
- [24] R. Oliveto, M. Gethers, D. Poshyvanyk, and A. De Lucia, “On the equivalence of information retrieval methods for automated traceability link recovery,” in Proc. of ICPC, 2010.
- [25] H. U. Asuncion, A. Asuncion, and R. N. Taylor, “Software traceability with topic modeling,” in Proc. of ICSE, 2010.
- [26] Gethers, Malcom , et al. "On integrating orthogonal information retrieval methods to improve traceability recovery," The College of Williams and." IEEE International Conference on Software Maintenance IEEE, 2011.
- [27] L. Hong and B. Davison. Empirical study of topic modeling in twitter. In Proceedings of the First Workshop on Social Media Analytics, pages 80–88. ACM, 2010.
- [28] X. Yan, J. Guo, Y. Lan, and X. Cheng, “A biterm topic model for short texts,” in Proc. 22nd Int. Conf. World Wide Web, 2013, pp. 1445–1456.
- [29] J. Chang and D. M. Blei, “Hierarchical relational models for document networks,” Annals of Applied Statistics, 2010.
- [30] Hayes J H , Dekhtyar A , Sundaram S K . Advancing Candidate Link Generation for Requirements Tracing: The Study of Methods.[M]. IEEE Press, 2006.
- [31] Wohlin, C., Runeson, P., H'ost, M., Ohlsson, M.C., Regnell, B., Wesslen, A. Experimentation in software engineering, Springer Science & Business Media, 2012.