International Journal of Recent Research and Review, Vol. XIII, Issue 3, September 2020
ISSN 2277 – 8322

Requirement Tracing Using Feature Selection Metric

Raghda Abdul-Jaleel Fathi
Department of Software Engineering, Mosul University, Iraq
raghda.jaleel@gmail.com

***Abstract* — It is substantial during the software life cycle to track the changes of requirements.During and at the end of each stage of the software life cycle, every requirement should be checked. In order to prove that the requirements have been met in the design, it is common to build Requirement traceability matrices (RTMs). In the various stages of software lifecycle, we need to create RTMs, but unfortunately a few developers and designers create RTMs.This paper demonstrates the applicability of feature selection metric to generate RTMs. It is applied and validated using two datasets and six types of filters (0, 0.05, 0.1, 0.15, 0.2, 0.25) for each dataset MODIS and CM1.This work aims to generate tracing links using the requirements tracing method. The main goal is to find a solution to the problem of tracing requirements by using the feature selection metric method (information gain) to improve the accuracy of the generated tracing links. The results show that the proposed method gives better result when compared with the traditional TF-IDF method.**

***Keywords* — Requirement Tracing; Information Retrieval; Feature Selection Metric; Information Gain.**

I. INTRODUCTION

The process of collecting requirements is the main stage in the project development. It is not possible for us to document the requirements, analyze, update or track them as the software development life cycle progresses [1].

The Verification and Validation (V&V) or Independent Verification and Validation (IV&V) analysts use tracing methods are the most common which works as follows. The first step,through the documents we extract the requirements. The analyst will perform a comparison between one of the high level requirement with the entire low level requirement. If he found that they are similar, the analyst will add this pair of (high-low level) to the link list. This in the case of tracing is from high to low level requirement [2].

The analyst is responsible for tracing the needs of stakeholder to the requirements model, and the test model. As for the designer, he is responsible for tracing requirements to the deliverables. Thus, analysts and designers are considered to be the main participants in the tracing requirements [3].

II. RELATED WORK

During the past years, many researchers offered their work in requirements tracing as follows:

In 2007 Hayes, Dekhtyar,Sundaram, Holbrook, and Vadlamudi argued how to automatically generate RTM, offered the results and matched them with manual generation of RTM[4].

Also in 2007Sundaramworked to improve the generated traceability links, various methods of Information Retrieval (IR) are applied and using user feedback. Wrinkles were also applied as filters to the main IR methods.The voting mechanism was used to choose the traceability links specified by various IR methods [2].

In 2009 Zou used three methods to develop and implement automated tracing tools and these methods are: (Query term coverage, Phrasing and Utilizing a project glossary). These three methods are for solving the low precision which is a lot to worry about especially in tracing dependent on the information retrieval [5].

In 2010, Cuddeback, Dekhtyar, Hayes research was based on a study from two universities and there were 26 analysts analyzing RTMs of different degrees of accuracy for the Java code format. This research depended on the performance of human analysts in examining RTMs [6].

In 2011sultanov, Hayes, kong focused on generating RTMs from high-level requirements to low-level requirements through the implementation of Swarm Intelligence which employs two algorithms (simple swarm and pheromone swarm).The results showed consistent or better accuracy than classic IR methods [7].

In 2012 Kong focused on how human analysts use tracing information generated by automated methods to create the final RTM and how to enhance the process of generating the RTM [8].

In 2014, Cleland-Huang, Gotel, Hayes, Mäder, Zisman identified seven studies in the traceability direction. Each direction of this research assists in the treatment to achieve traceability. Each part of this research it is fully implementable as it helps our organization to work cooperatively to develop traceability[9].

In 2016, Rick Kok introduced more advanced traceability mechanisms. By doing an experiment using data from production environment. Researchers have found that there is likely to be an increase implementation performance in the long-range at the same time leading to improved efficiency and reduce effort in requirements management[10].

In 2018, Dawood and Sahraoui built a structure to generate diagrams of design from requirements in a semi-automatic way and built traceability between requirements and design phases, and in contrast. This structure explains how to manage traceability in various levels, and how to implement these changes[11].

III. REQUIREMENTTRACING

When requirements being traced from high to low level, tracing is defined as the ability to track requirements in specifications to their source in a group of documented links [12].

Requirements Tracing is done in the following format:

First step, parsed and tokenized is done for each requirement. Then stopwords is done to delete all words that are not beneficial in retrieval,*for example* (“Shall”, “the”, “for”). In order to guarantee that the other formats of the same word can be treated as one term (e.g., “information” and “informational”), so the stemmed will be applied to the remaining tokens. Finally, create and store for the document vector representation [13].

In fact, the retrieval process is as following: After the requirements vectors are created, for the case of forward-tracing, the list of queries, which represent high-level requirements, will be processed as a query after the other, and it will also be converted into query vectors.Then similarity will be calculated for each pair of query-design element, then for each query, a list of design elements with non-zero similarity values will be generated, then similarity values are arranged in descending order. These lists are returned to the analyst [13].

IV. INFORMATION RETRIEVAL

Information retrieval (IR) is a search in a set of documents for a document that is relevant to a specific information. In a similar outline, the weight relevance arranges the returned documents in response to a query. It is a value in counted numbers that indicates how accurately the returned document matches the query. The document that is higher relevant to the query means that is a greater weight. From the user's point of view, the document is relevant if he thinks that the document is relevant to the original query. He may not accept with the high weight value that appears for each retrieved document [14].

Initially the behavior of traditional IR methods was applied to requirements tracing problems.The methods that were applied were as follows[15]:

The vector of information retrieval, called *standard vector model* (also known as *tf-idf model*) is defined as follows [16,17]:

Let V = {k_1,..., k_N) be the vocabulary of a given document collection. Then, a vector model of a document d is a vector (w_1, ..., w_N) of keyword weights, where w_i is computed as Eq. (1)[2,16,17]:

$$\boldsymbol{w_i = tf_i\,(d)\,.\,idf_i} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Where tfi(d) is the term frequency of the ith keyword in document d , idf_i is the inverse document frequency of the ith term in the document collectionis computed asEq. (2)[2,16,17]:

$$\boldsymbol{idf_i = log_2 \frac{n}{df_i}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where df_i is the total number of documents containing the ith term in the document collection, and n is the size of the document collection.

Given a document vector d=(w_1,… , w_N) and a

similarly computed query vector q=(q_1,…, q_N) the similarity between d and q is defined as the cosine of the angle between the vectors asEq. (3)[2,16,17]:

$$\boldsymbol{sim(d,q) = \cos(d,q) = \frac{\sum_{i=1}^{N} w_i \cdot q_i}{\sqrt{\sum_{i=1}^{N} w_i^2 \cdot \sum_{i=1}^{N} q_i^2}}} \ldots\ldots (3)$$

V. EMPLOYD FILTERS

To generate the list of candidate links with relevance higher than one of the predefined levels, six filters were used: 0, 0.05, 0.1, 0.15, 0.2, and 0.25. These filters evaluate the quality of the candidate link list.

VI. MEASURING THE EFFICIENCY

We designed a requirement tracing tool to time saving of the analyst as well as generating RTMs with high quality, in addition to improving the value of (*Recall*) and (*Precision*).

When calculating two metrics, can give us an evaluation of requirements tracing as follows [13]:

Recall is the actual match rate that can be found, as Eq.(4) [13].

$$\boldsymbol{Recall = \frac{Number\ of\ matches\ found\ by\ the\ IR\ method}{Total\ number\ of\ possible\ matches}} \ldots.(4)$$

Precision is the percentage of all candidate links that returned because it considers the correct matches of these links, as Eq. (5) [13].

$$\boldsymbol{Precision = \frac{Number\ of\ true\ links\ found}{Total\ number\ of\ links\ returned\ by\ the\ IR\ method}} \ldots.(5)$$

VI. TEXT CATEGORIZATION

The operation of arranging text documents in one or more predetermined categories or classes of similar documents is called *Text categorization*. A set of features are selected to demonstrate the combination of a specific document with a specific category. The set of features creates differences in the results of this categorization. The method of arranging text documents into similar categories reduces the overhead for a quick retrieval of these documents and provides a smaller domain in which the user can explore a similar document, this thing is realized by the advocates of text categorization[18].

Supervised learning is in-demand in most of the real-world classification problems. It is not known in both main class probabilities and class-conditional probabilities. Subsequently, In order for this field to be represented in the best way, many candidate features have been introduced. Regrettably, many of these are either partially or totally irrelevant/redundant to the desired concept. There are three features in relation to the desired concept which are relevant, irrelevant and redundant. A *relevant feature* which is must be neither irrelevant nor redundant to the desired concept, an *irrelevant feature* does not affect the desired concept in any way, and a *redundant feature* does not add anything new the desired concept. In multiple applications, sometimes learning will not work well before removing these undesirable features especially when the volume of dataset is so huge. In order to significantly reduce the running time of the learning algorithm and give a more general concept, we must reduce the number of irrelevant/redundant features. This will help in understanding more the concept of real-world classification problems. Feature selection styles are trying to choose a subset of relevant features to the desired concept[19].

VII. FEATURE SELECTION METRIC

By reducing noise, which is also known as reducing feature space dimensionality and removing non-informative features, the information for each document given to the classifier will enhance the quality of the related content.

Therefore, reducing feature space dimensionality is a critical stage on the whole results. For this purpose, a well-known feature selections metric is selected: Information Gain [20].

VIII. INFORMATION GAIN

IG is a substantial feature selection method that calculates how much the feature is informative about the class. IG explain the unconfirmed reduction in

identifying category by defining the value of the feature. IG is a classification points method which can be counted for a term by as Eq.6 [21]:

As shown in the following figure (Fig.1),there is a horizontal line divided into two categories:above the line is the positive category and below the line is negative category. Each column represents the document distribution within the corpus for each term ($t_1, t_2, t_3, t_4, t_5 and\ t_6$). The height in one column represents the number of documents in the corpus. The number of documents containing this term is in height of the shaded part. To declare the number of different documents, we use a, b, c, and d as follow [22]:

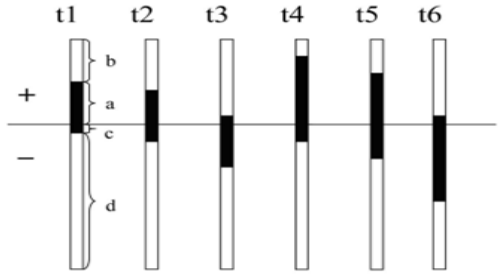

Figure 1Example of different distributions of documents that contain six terms in the whole collection

Where [22]:

a is the number of documents in the positive category that contain this term.

b is the number of documents in the positive category that do not contain this term.

c is the number of documents in the negative category
that contain this term.

d is the number of documents in the negative category that do not contain this term.

$$ig = \frac{a}{N} * log \frac{a * N}{(a+c) * (a+b)} + \frac{b}{N} * log \frac{b * N}{(b+d) * (a+b)} + \frac{c}{N} * log \frac{c * N}{(a+c) * (c+d)} + \frac{d}{N} * log \frac{d * N}{(b+d) * (c+d)} \quad ... (6)$$

IX. DATA SETS

This work is validated using two NASA open source datasets. Both MODIS and CM-1 datasets are used here to assess the utilized techniques of IR. The MODIS dataset consists of 19 high level and 49 low-level requirements, where the CM-1 dataset contains 235 high-level requirements and 220 design elements. A manual tracing was done for both datasets for verification; these are referred to as "answer sets" or "theoretical true traces". There were 41 and 361 true links found for the MODIS and CM-1 datasets, respectively.

X. EXPERIMENTAL RESULTS

Feature Selection Metric is presented in this paper, along with a discussion of the preprocessing techniques that are commonly used. First, the documents are parsed using two approaches as follow: *Statistical Format*, this method represents all the texts in high and low level requirements and *Linguistic Parser*, this method is done by entering the texts in the high and low-level requirements into the parser and analyzing the texts. Only four types of parser outputs are taken according to the (POS) tagger, and these outputs are (NN, NNS, NNP, NNPS). Whether it is a statistical or linguistic process, Stem process will be applied, and the stem process will be carried out either by (Porter Stemming Algorithm) or by the proposed method, which is (Dictionary Stemming). In this work, it is suggested to build a dictionary, due to the various problems in the Porter algorithm and the lack of a specialized dictionary for the Stemming work. Finally, the term frequency is computed using information gain metric rather than TF-IDF. In this paper the vector space model is used for Information Retrieval.

The six filters were used together with the metrics described previously using MODIS and CM1 datasets. The results are compared with those found in [23].

A. First Dataset (MODIS) with Filters (0, 0.05, 0.1, 0.15, 0.2 and 0.25):

In this section, experiments are done using the MODIS Dataset and filters (0, 0.05, 0.1, 0.15, 0.2 and 0.25). Table (I) and (II) show the results of running the Information Gain for each filter. The results for all filter show that the

values of Recall and Precision for all filter improved except those marked with a (*).

Table I
RESULT OF INFORMATION GAIN IN MODIS DATASET WITH PORTER ALGORITHM

Format	Filter	Recall	Precision
XML	0	79.6	7.9
Statistical		75.6*	9.0
Linguistic		70.7*	9.6
XML	0.05	48.7	7.7
Statistical		75.6	10.2
Linguistic		70.7	10.0
XML	0.1	29.2	11.7
Statistical		58.5	17.3
Linguistic		68.2	14.6
XML	0.15	24.4	17.2
Statistical		34.1	17.5
Linguistic		53.6	19.8
XML	0.2	19.5	21.6
Statistical		29.2	34.2
Linguistic		41.4	22.0
XML	0.25	19.5	32.0
Statistical		26.8	45.8
Linguistic		29.2	24.0*

Table II
RESULT OF INFORMATION GAIN IN MODIS DATASET WITH DICTIONARY

Format	Filter	Recall	Precision
XML	0	79.6	7.9
Statistical		75.6*	9.2
Linguistic		70.7*	9.6
XML	0.05	48.7	7.7
Statistical		75.6	10.4
Linguistic		70.7	9.8
XML	0.1	29.2	11.7
Statistical		58.5	17.1
Linguistic		68.2	14.3
XML	0.15	24.4	17.2
Statistical		34.1	17.5
Linguistic		53.6	19.6
XML	0.2	19.5	21.6
Statistical		29.2	35.2
Linguistic		41.4	22.0
XML	0.25	19.5	32.0
Statistical		26.8	47.8
Linguistic		29.2	24.0*

B. First Dataset (CM1) with Filters (0, 0.05, 0.1, 0.15, 0.2 and 0.25):

In this section, experiments are done using the CM1Dataset and filters (0, 0.05, 0.1, 0.15, 0.2 and 0.25). Table (III) and (IV) show the results of running the Information Gain for each filter. The results showed that most of the Recall values that have been written in bold are better than the values compared with them. As for the values of Precision, one value that have been written in bold in table III is better than the values compared with them.

Table III
RESULT OF INFORMATION GAIN IN CM1 DATASET WITH PORTER ALGORITHM

Format	Filter	Recall	Precision
XML	0	97.8	1.5
Statistical		**98.6**	1.0
Linguistic		89.4	**1.6**
XML	0.05	92.2	4.3
Statistical		**95.2**	1.4
Linguistic		85.5	2.1
XML	0.1	76.4	10.8
Statistical		**89.1**	2.6
Linguistic		73.4	3.1
XML	0.15	53.7	19.1
Statistical		**78.3**	4.4
Linguistic		**62.6**	4.7
XML	0.2	32.6	27.1
Statistical		**67.5**	6.8
Linguistic		**50.1**	6.5
XML	0.25	21.9	34.8
Statistical		**53.7**	10.4
Linguistic		**39.0**	9.3

Table IV
RESULT OF INFORMATION GAIN IN CM1 DATASET WITH DICTIONARY

Format	Filter	Recall	Precision
XML	0	97.8	1.5
Statistical		**98.6**	1.0
Linguistic		**98.6**	1.0
XML	0.05	92.2	4.3
Statistical		**95.2**	1.4
Linguistic		**95.2**	1.4
XML	0.1	76.4	10.8
Statistical		**88.9**	2.5
Linguistic		73.1	3.1
XML	0.15	53.7	19.1
Statistical		**78.1**	4.2
Linguistic		**61.4**	4.6
XML	0.2	32.6	27.1
Statistical		**67.8**	6.7
Linguistic		**49.8**	6.4
XML	0.25	21.9	34.8
Statistical		**53.4**	10.1
Linguistic		**38.5**	9.0

XI. CONCLUSIONS AND FUTURE WORK

In this paper, the effectiveness of information retrieval methods in automating the tracing of textual requirements was examined. The information gain was evaluated and it was found that there were better results for Recall and Precision compared to TF-IDF. In this work, the vector space model was adapted for information gain, in addition to the Statistical and Linguistic format. Porter Stemming Algorithm and Dictionary for stemming were applied using two open source datasets (MODIS and CM1).Future work can carry on in several directions, such as the use of another technique in Information Retrieval (IR), as well as the vector space model and another Feature Selection Metric rather than information gain. More methods can be sought to be employed other than to Feature Selection Metric enhance results.

XII. REFERENCES

[1] D.K. Susan Geethu ,R. Subha ,S. Palaniswami, *Semantic Swarm Intelligence for Candidate Links*, In International Journal on Cybernetics & Informatics (IJCI) Vol.1, No.3, June 2012.

[2] S.K. Sundaram, *Requirements Tracing Using Information Retrieval*, Dissertation for the Degree of Doctor in Philosophy (Ph.D.) in Engineering , University of Kentucky, 2007.

[3] G.A. Stout, *Requirements Traceability and the Effect on the System Development Lifecycle (SDLC)*, Technical Report, DISS 725, Spring Cluster, 2001.

[4] J.H. Hayes, A. Dekhtyar, S.K. Sundaram, A. Holbrook, S. Vadlamudi, A. April, *REquirements Tracing On target (RETRO):Improving Software Maintenance through Traceability Recovery*. In Innovations in Systems and Software Engineering 3 (3), pp:193-202,2007.

[5] X. Zou, *Improving Automated Requirements Trace Retrieval Through Term-Based Enhancement Strategies*, Dissertation for the Degree of Doctor in Philosophy (Ph.D.), College of Computing and Digital Media, DePaul University, 2009.

[6] D. Cuddeback, A. Dekhtyar, J.H. Hayes, *Automated Requirements Traceability: the Study of Human Analysts*, Requirements Engineering Conference (RE), 2010 18th IEEE International, Sept. 27 2010-Oct. 1 2010, pp: 231-240 , ISSN :1090705X , Print ISBN:978-1-4244-8022-7.

[7] H. Sultanov, J.H. Hayes, W. Kong, *Application of swarm techniques to requirements tracing*, Journal of Springer-Verlag London Limited 2011, N0:10, 2011.

[8] W. Kong, *Improving Traceability Recovery Techniques Through The Study Of Tracing Methods And Analyst Behavior*, Dissertation for the Degree of Doctor in Philosophy (Ph.D.) in Engineering , University of Kentucky, 2012.

[9] J. Cleland-Huang, O. Gotel,J.H. Hayes,P. Mäder, A. Zisman, *Software Traceability: Trends and Future Directions*, the 36th International Conference on Software Engineering (ICSE), Hyderabad, India, 2014.

[10] R. Kok ,*Tracing Requirements in an Insurance Software Development Company*, Dissertation for the Degree of Master in Business Informatics Department of Information and Computing Sciences, Utrecht University, July 2016.

[11] O. S. Dawood, A.Sahraoui, *Toward Requirements and Design Traceability using Natural Language Processing*, European Journal of Engineering Research and Science (EJERS), Vol. 3, No. 7, July 2018.

[12] Ali M.J., *Metrics for Requirements Engineering*, Dissertation for the Degree of Master of Science (M.Sc.) in Computing Science, Umea University,2006.

[13] J.H. Haye, A. Dekhtyar, J. Osborne, *Improving requirements tracing via information retrieval*. In Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International, 8-12 Sept. 2003, pp:138-147. ISSN :1090-705X, Print ISBN:0-7695-1980-6.

[14] H. Sultanov , *Application Of Swarm And Reinforcement Learning Techniques To Requirements*, Dissertation for the Degree of Doctor in Philosophy (Ph.D.) in Engineering , University of Kentucky,2013.

[15] J. Hayes, A. Dekhtya, *Final Report for Robust Requirements Tracing via Internet Search Technology: Improving an IV&V Technique - Phase II*, 2/15/03 – 7/31/04, $86K, NASA IV&V Facility and GSFC.

[16] J.H. Hayes, A. Dekhtyar, S.K. Sundaram, S. Howard, *Helping Analysts Trace Requirements: An Objective Look*. In Requirements Engineering Conference. Proceedings. 12th IEEE International, 6-11 Sept. 2004 pp: 249 – 259. ISSN :1090-705X. Print ISBN:0-7695-21746,2004.

[17] J.H.Hayes, A. Dekhtyar, S.K. Sundaram, S. Howard,*On Effectiveness of User Feedback-based Information Retrieval Methods for Requirements Tracing*, (TR 423-04),2004.

[18] A. Basu, C.Watters, M. Shepherd, *Support Vector Machines for Text Categorization* , Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03), Hawaii, USA. Washington: IEEE Computer Society, 2003.

[19] M. Dash, H. Liu, *Feature Selection for Classification*, In Intelligent Data Analysis , vol .1, pp:131-156, 1997.

[20] F. Peleja, G.P. Lopes, J. Silva, *Text Categorization: A comparison of classifiers, feature selection metrics and document representation*, Proceedings of the 15th Portuguese Conference in Artificial Intelligence , pp:660-674. ISBN: 978-989-95618-4-7,2011.

[21] O.Al-Harbil, *A Comparative Study of Feature Selection Methods for Dialectal Arabic Sentiment Classification Using Support Vector Machine*, International Journal of Computer Science and Network Security(IJCSNS), VOL.19 No.1, January 2019.

[22] M. Lan, C.l. Tan, J. Su, Y. Lu ,*Supervised and Traditional Term Weighting Methods for Automatic Text Categorization*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 31, NO. 4,Pages:721-735. ISSN :0162-8828,2009.

[23] S.K.Sundaram, J.H.Hayes, A.Dekhtyar,*Baselines in Requirements Tracing*. In ACM SIGSOFT Software Engineering Notes 30 (4), pp:1-6,2004.