Semi-supervised Approach for Recovering Traceability Links in Complex Systems

Emma Effa Bella
Sorbonne Université
CNRS- LIP6
IRT SystemX
Paris, France
emma-effa.bella@irt-systemx.fr

Marie-Pierre Gervais
Université Paris Nanterre
CNRS- LIP6
Paris, France
marie-pierre.gervais@lip6.fr

Reda Bendraou
Université Paris Nanterre
CNRS- LIP6
Paris, France
reda.bendraou@lip6.fr

Laurent Wouters
Association Cénotélie
Paris, France
lwouters@cenotelie.fr

Ali Koudri
IRT SystemX
Paris, France
ali.koudri@irt-systemx.fr

***Abstract*— Building a complex system requires the collaboration of different stakeholders. They work together to model the system keeping in mind the requirements described in specification documents. This complexity induces a large volume of requirements and models, i.e., artefacts that will be subject to frequent changes during the project lifetime. Since the artefacts are correlated with each other's, each change has to be rigorously propagated. Identifying traceability links between system's artefacts is then a critical step to reach this goal. In Information Retrieval domain, many approaches have been already proposed to cope with traceability issues. Their main drawback is they introduce an important amount of false positive links making the traceability links validation phase time consuming and error-prone. In this paper, we propose an approach that identifies traceability links with a reduced amount of false positive links ranging from 20% to 30% while raising the amount of true links identified up to 70%. The approach consists of three main steps: 1) we measure syntactical and semantic similarities between pair of artefacts by combining the use of four major Information Retrieval techniques; 2) using these similarity measures, we identify the most likely true and false links and we build the so called training data set; 3) this training data set and the four IR techniques are used as input of a predictive model in order to classify between true and false links leading ultimately to a reduced amount of false positives. The output is given in the form of a confidence measure that will help the modeller validating the traceability links. We evaluated our approach using four well-known public case studies. Each one comes with a clear identification of true traceability links which allowed us to compare with the outcome of our approach and validate its effectiveness.**

***Keywords*—traceability, information retrieval, requirements, models, complex systems**

I. Introduction

The development of complex systems involves the collaboration of many stakeholders. In order to design the system, they produce many artefacts i.e., requirements and models that are correlated with each other's and which evolve constantly. In such a volatile environment, there is a critical need to manage the impact of the different changes occurring during the project lifetime. Traceability, as defined by Edwards and Howell [1], is "A technique used to provide a relationship between the requirements, the design and the final implementation of the system." In complex systems engineering, establishing such traceability involves dealing with a large volume of requirements and models [2]. For example, the full specification of an aircraft includes about 10,000 requirements and a subway line of about 6,000. And modelling an aircraft can lead to hundreds of thousands of elements in hundreds of different models. Dealing manually with traceability issues in such a context is obviously unbearable.

In the literature, many works propose to automate the identification of traceability links. In the Information Retrieval (IR) community, approaches such as Vector Space Model (VSM), Latent Semantic Indexing (LSI), and Latent Dirichlet Allocation (LDA) have been used to recover traceability links between artefacts. However, traceability identification in these approaches is still complex and error prone [3]. Due to that limited accuracy, candidate links are systematically checked by an analyst which manually classifies them into two groups: the approved links called true links and the rejected ones called false links. Thereby, the candidate links evaluation is itself labour intensive and time-consuming [4] due to the significant number of false links that represent 90% of the candidate links at a low threshold.

Our approach aims at improving the evaluation of candidate links process by reducing the number of false links retrieved by IR techniques while maintaining a significant number of true links. It belongs to the semi-supervised approaches category as it makes use of links that are not yet validated in conjunction with a small amount of training set. However, in our industrial context, when we start a development project of a complex system, such a training set does not exist. Therefore, our proposal includes the definition of a strategy to build a small training data set.

The approach consists of 3 phases. First, we combine syntactical and semantics similarities measures of four IR techniques, i.e. LSI [7], VSM [5, 8], LDA [19] and word embedding [10]. Second, we use these similarities measures to build a small training data set. Finally, this training data set and the four IR techniques are used as input of a predictive model [17] in order to classify true and false links.

We evaluate our approach by investigating four open-source datasets. The results are very promising and show a reduced amount of false positive links ranging from 20% to 30% while maintaining the amount of true links up to 70%.

The remainder of this paper is organized as follows. Section II describes the overview of the approach. Section III presents the results of a preliminary evaluation of the approach on four public case studies. Section IV discusses related works. Finally, Section V concludes the paper and describes directions for future work.

II. Overview of the Approach

Fig. 1 presents an overview of our approach for identifying traceability links between artefacts, namely requirements and models. This is based on the combined use of four well-known IR techniques, LSI, LDA, VSM and Word embedding. IR techniques are used to provide similarity measures between pairs of textual software engineering artefacts. Similarity measures are essential to solve pattern classification problems. LSI, LDA and VSM have been intensively used to provide support for several software engineering tasks, and more recently, word embedding has been successfully employed in various natural language processing tasks [14, 15]. Word embedding represents words as vectors of real numbers that capture their contextual semantic meanings.

We illustrate our approach with an example from the Icebreaker dataset, which is one of the dataset used in the Evaluation Section. In this excerpt, there are one requirement R1 "*Historical forecast data shall be retrieved as needed*" and one class "*Weather Forecast GUI*" whose attribute is *date* and operations are *set Date()*, *get Date()*, *get Forecast-History()*, *handle-Critical-Forecast()* *and*, *get Forecast-Input()*.

Our approach takes as input requirements document described in natural language and models in XML Metadata Interchange (XMI) format. In Step 1 "C*ompute similarities"*, we perform pre-processing of the input data which consists of the tokenization and Stopwords removal. Stopwords [12] are tokens that appear so frequently in the artefacts that they become irrelevant for link recovery. This pre-processing is achieved by means of Natural Processing Language functionalities embedded into the IR techniques we use for quantifying similarities between each pair of artefacts. These measurements are based on morphological similarities between terms contained in the artefacts (LSI, VSM), topics proximity of artefacts (LDA) or semantic proximity of terms (word embedding [14, 15]). We then use Word Mover's Distance (WMD) [15] which assesses the distance between a pair of artefacts using word embedding [14]. Thus, each similarity measure provides a description of each pair of artefacts according to a given criterion. A vector containing similarity scores is constructed for each pair of potentially related artefacts. As a result, the more the similarity measures are diverse and complementary, the more this vector will contain enough information to decide whether a link exists or not between two given artefacts. Thus, for a pair of artefacts *(x, y)* with $(m_1, m_2 ..., m_p)$ the similarity measures assigned by the *p* IR techniques, the associated descriptor vector would be equal to $(m_1(x, y), m_2(x, y)..., m_p(x, y))$. Each descriptor vector constitutes a row of the descriptor matrix, which is the output of step 1.

In our example, the descriptor vector of the pair of artefacts R1 and the *Weather Forecast GUI* class with LSI, VSM, LDA and WMD measures is $((0.94)_{LSI}, (0.44)_{VSM}, (0.71)_{LDA}, (1.03)_{WMD})$.

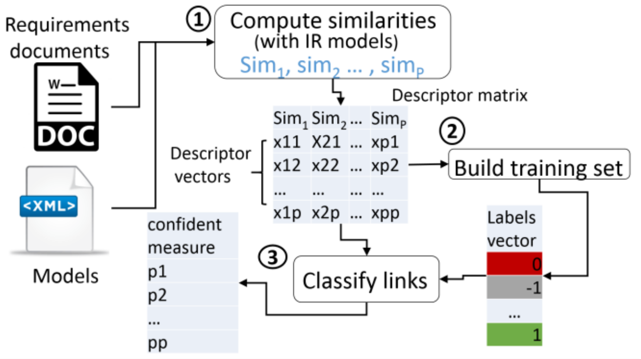

Fig. 1. Overview of our 3 steps approach

If a set of validated links would be available, one could use a supervised classification method [16] to derive a statistical model that estimates the probability for two artefacts to be linked, based on their descriptor vector. This probability could constitute a measure of confidence on the traceability link between two artefacts. Indeed, deep learning approaches have recently achieved outstanding performance in the traceability community [10]. However as mentioned earlier, in our industrial context, a training dataset, i.e. a set of artefacts with validated links, is generally not available. In step 2 "*Build training set"*, we propose a heuristic to label some pairs of artefacts as related or non-related. This heuristic is motivated by the statistical ranking of IR techniques [18], which puts the most likely links on the top and the bad links (false links) to the bottom. Thus, for each similarity measure, we take the pairs of artefacts with a small percentage of the highest and the lowest similarity scores. In the Evaluation Section, we use empirically the 10% highest and the 10% lowest similarity scores respectively. The list of links labelled as true is obtained as the *intersection* of the set of links having the highest scores and the list of links labelled as false is obtained by the *union* of the set of links with the lowest scores. At the end of this step, the labels vector is built. Potential true links are represented with the value "1", the false links are represented with the value "0" and the remaining links, called unlabelled links, are represented with the value "-1". In our example, the labels vector belongs to the unlabelled links because of the disparities of the similarity measures therefore it has the value -1.

In step 3 "C*lassify Links"*, we use the descriptor matrix and the labels vector as inputs. Based on these inputs, a predictive model finds the community structures in order to group them in true and false links. Given the important amount of unlabelled data, in order to build our predictive model, we use the label spreading method, which is a semi-supervised machine learning technique. We use label spreading [17] among other semi-supervised learning methods because of its ability to modify initially provided labels and therefore, its potential robustness to labelling errors. The label propagation algorithm and its variant the label spreading are used for community detection in large complex networks. Therefore, label spreading efficiency for our problem relies on the clustering hypothesis, which is the assumption that true links tend to gather in the descriptors space [8]. As output, the predictive model provides a probability which describes how much a link belongs to the class of true links. As previously stated, we will refer to this probability as a confidence measure. From a practical point of view, it provides a prioritization criterion to the analyst during the links verification phase. In our example, the confidence measure of the requirement R1 and the *Weather Forecast GUI* class is *0.19*.

III. EVALUATION

The experiments for the validation of our approach have been conducted over four datasets available on the COEST[1] website: Icebreaker, HIPAA (Healthcare Insurance Portability and Accountability Act), EasyClinic, and CM1-NASA (NASA spacecraft instrument). The Icebreaker dataset was built by students. It provides traceability links between High-level requirements and UML Classes. The HIPAA provides traceability links between 10 HIPAA Technical safeguards and requirements in 10 different requirements specifications. The CM1-NASA is a dataset which provides high-level requirements and low-level requirements documents. The EasyClinic dataset was also built by a small group of students. It contains several artefacts including use cases, interaction diagrams, test cases and classes description.

VSM has been implemented via the Tracelab tool [13]. LSI, LDA and WMD have been implemented with Gensim[2]. We experimented all the possible combinations of the four IR techniques and the best one in the classification problem was: VSM-LSI-LDA. We compare the performance of this combination against the performance of each individual IR technique. The most used metrics for evaluating any IR techniques are recall, precision and F-measure. F-measure is the harmonic average of the recall and the precision. It shows trade-offs between precision and recall. It can then be used to provide insights about the performance of a method. A pair of artifacts is considered related when the similarity measure is greater or equal to a given threshold. The accuracy rate of IR technique strongly relies on this threshold. Thus, we evaluated the results by plotting F-measure at different thresholds for each method on the four datasets. The F-measure results of all methods in Fig. 2, Fig. 3, Fig. 4, and Fig. 5, show that our approach is more effective than VSM, LSI, and LDA at low threshold from 0 to 0.3. Precisions at low thresholds are significantly improved by our approach; especially at low threshold from 0 to 0.3. Our precision is much higher than LSI, VSM and WMD. Concerning to LDA, it depends on the chosen threshold. The higher the threshold, the better LDA is compared to our approach. Concerning the recall, our approach is lower at high thresholds from 0.3 to 0.9. Nevertheless, our approach achieves reasonably high recall values, from 50% to 70% at low threshold points from 0 to 0.3 in all datasets.

In summary, our preliminary experimental results demonstrate that our approach improves precision at lower threshold from 0 to 0.3 while keeping a high recall. In other words, this significantly decreases the false links at lower threshold from 0 to 0.3 ranging from 20% to 30% of the candidate links while raising the amount of true links up to 70%. Its main limitation is that the confidence values provided by the predictive model are lower; typically, the

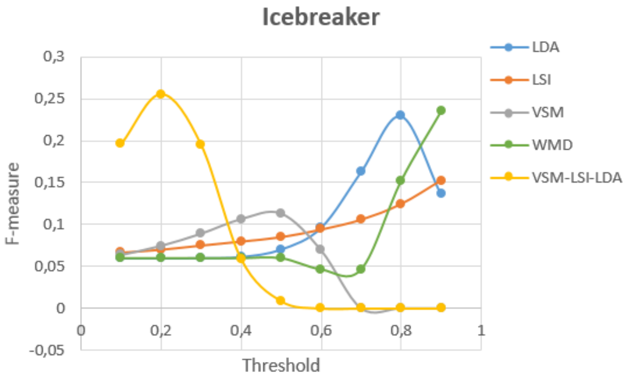

Fig. 2. F –measure results for Icebreaker dataset

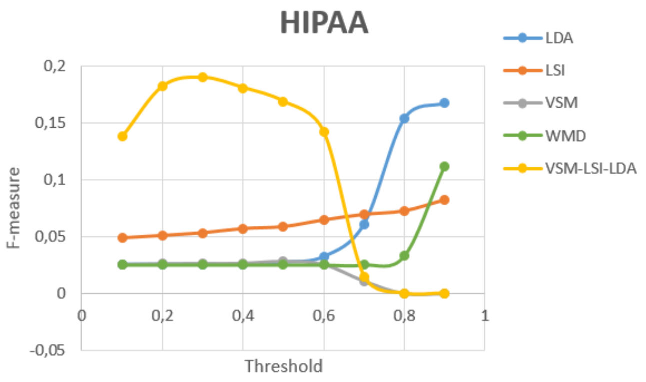

Fig. 3. F –measure results for HIPAA dataset

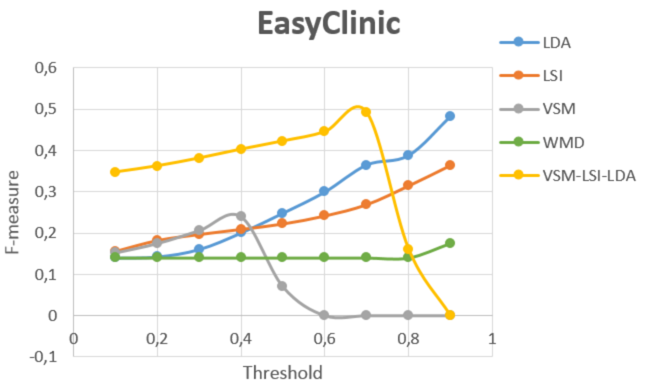

Fig. 4. F –measure results for Easyclinic dataset

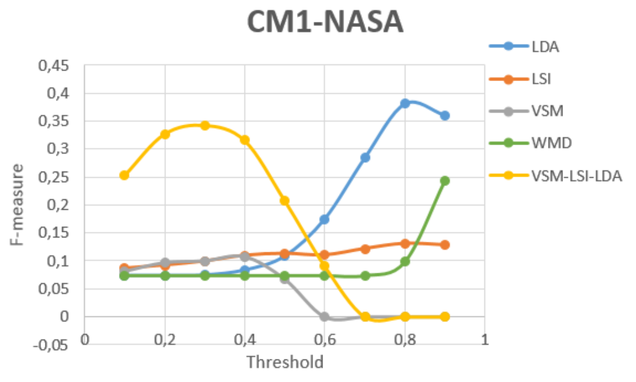

Fig. 5. F –measure results for CM1-NASA dataset

[1] Center of Excellence for Software & Systems Traceability, http://www.coest.org/

[2] https://radimrehurek.com/gensim/index.html

class of true links is from 0.1 to 0.5. Therefore, at higher threshold, the number of retrieved links is very low. This is because the predictive model can effectively distinguish true and false links at low threshold, due to VSM, LSI, and LDA which achieve recall levels above 90% at lower thresholds.

IV. RELATED WORKS

Approaches combining different techniques have been proposed in the literature to improve the IR techniques and overcome their limits. For example, Chen et al. [5] combine Regular Expression, Key Phrases, and Clustering algorithm K-mean to enhance the performance of VSM. Their approach has higher precision and recovers more true links than VSM alone. Cleland-Huang et al. [6] propose three enhancement strategies (hierarchical modelling, logical clustering of artefacts, and semi-automated pruning of the probabilistic network (PN)) to improve the performance of PN. The results indicate that these strategies effectively improve trace retrieval performance. Wang et al. [7] present four strategies (source code clustering, identifier classifying, similarity thesaurus, and hierarchical structure enhancement) to improve LSI. Their approach has higher precision but lower recall. Our work differs from the above in that it combines IR techniques in order to improving their effectiveness while compensating for their weaknesses.

Our proposed approach is most related to predictive models which have been applied to identification of traceability links. For instance, Mills et al. [9] use a predictive model with two features (text retrieval rankings and query quality metrics) to automatically classify links as true or false. Their approach achieves high accuracy on average using both types of features but there are still a high number of miss-classified links. Many researchers have found that unlabeled data, when used in conjunction with a small amount of labelled data, can produce considerable improvement in learning accuracy [11]. The novelty of this work is that quantitative similarity measures have been considered as features to build a classification model. Our main goal is to reduce the number of false links and consequently reduce potential errors due to manual candidate links evaluation.

V. CONCLUSION

In this paper, we propose a semi-supervised method, which uses a combination of similarities measure defined by IR techniques in order to classify true links and false links. Due to unavailability of validated data training, we define a strategy to provide a small dataset. The preliminary results on four datasets indicate that our approach is more accurate on average by 3-5% than LDA and LSI and by 10% than VSM and WMD. In future work, we plan to investigate more strategies to build data training set as well as different configurations of the semi-supervised method, which could improve the performance of our approach. An ongoing validation is in progress with our industrial partners. Moreover, this approach is the first step to reach our ultimate goal, which is the maintenance of links during the project lifetime. In summary, the presented results in this paper demonstrate that semi-supervised techniques can be effectively applied to traceability links recovery.

REFERENCES

[1] M. Edwards and S. L. Howell, “A methodology for systems requirements specification and traceability for large real time complex systems,” No. NAVSWC-TR-91-584. NAVAL SURFACE WARFARE CENTER SILVER SPRING MD, 1991.

[2] O. Gotel, et al. "The grand challenge of traceability (v1. 0)." Software and Systems Traceability. Springer London, pp 343-409. 2012.

[3] M. Heindl and S. Biffl, "A case study on value-based requirements tracing," In Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering. ACM, 2005.

[4] D. Cuddeback, A. Dekhtyar, J. H. Hayes, J. Holden and W. K. Kong, "Towards overcoming human analyst fallibility in the requirements tracing process (NIER Track)," In Proceedings of the 33rd International Conference on Software Engineering (ICSE), may 2011.

[5] X. Chen, J. Hosking, and J. Grundy. "A combination approach for enhancing automated traceability:(NIER track)." In the 33rd International Conference on Software Engineering (ICSE), may 2011.

[6] J. Cleland-Huang, R. Settimi, C. Duan and X. Zou, "Utilizing supporting evidence to improve dynamic requirements traceability." In Proceedings of the 13th IEEE International Conference on Requirements Engineering, pp. 135-144, August 2005.

[7] X. Wang, G. Lai, and C. Liu, “Recovering relationships between documentation and source code based on the characteristics of software engineering,” Electronic Notes in Theoretical Computer Science, vol. 243, pp. 121–137, 2009.

[8] N. Niu, and A. Mahmoud. "Enhancing candidate link generation for requirements tracing: the cluster hypothesis revisited." In Proceeding of the 20th International Conference on Requirements Engineering (RE). IEEE , pp. 81-90, september 2012.

[9] C. Mills, and S. Haiduc. "A machine learning approach for determining the validity of traceability links," In the 39th International Conference on Software Engineering Companion (ICSE-C), IEEE/ACM , 2017.

[10] J. Guo, J. Cheng, and J. Cleland-Huang. "Semantically enhanced software traceability using deep learning techniques." Proceedings of the 39th International Conference on Software Engineering. IEEE Press, 2017.

[11] X. Zhu, “Semi-supervised learning literature survey”. Computer Sciences, University of Wisconsin-Madison, 2008.

[12] C. D. Manning, P. Raghavan, and H. Schütze. “Introduction to information retrieval (Vol. 39)”. Cambridge university press, 2008

[13] J. Cleland-Huang, et al. "Grand challenges, benchmarks, and TraceLab: developing infrastructure for the software traceability research community." In Proceedings of the 6th international workshop on traceability in emerging forms of software engineering. ACM, pp. 17-23, 2011.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality." In Advances in neural information processing systems. pp. 3111-3119, 2013

[15] M. Kusner, Y. Sun, N. Kolkin and K. Weinberger. "From word embeddings to document distances." In International Conference on Machine Learning. pp. 957-966, 2015.

[16] T. M. Mitchell, "Machine learning and data mining." Communications of the ACM 42.11 pp : 30-36, 1999.

[17] O. Delalleau, Y. Bengio, and N. Le Roux. "Efficient Non-Parametric Function Induction in Semi-Supervised Learning." In AISTATS. (Vol. 27. No. 28. pp. 100), 2005.

[18] W. B. Frakes, and R. Baeza-Yates, (eds.) " Information retrieval: Data structures & algorithms, (Vol. 331)". Englewood Cliffs, New Jersey: prentice Hall, 1992.

[19] A. Panichella, et al. "How to effectively use topic models for software engineering tasks? an approach based on genetic algorithms." In proceedings of the 35th International Conference on Software Engineering (ICSE). IEEE, pp. 522-531 2013.