SECTION: TECHNICAL SCIENCES

DOI 10.70286/isu-13.08.2025.008

GRAPH EMBEDDING AS A FORMAT FOR SOFTWARE RISKS REPRESENTATION

Lyashkevych Mariia
Assistant Professor
Shuvar Roman
Candidate of Physical and Mathematical Sciences, docent
Head of System Design Department
Department of System Design
Ivan Franko National University of Lviv, Ukraine

Software development requirements, like the business environments, are changing rapidly due to market conditions. Therefore, it is important to have a good understanding of the accuracy and completeness of software requirements. Low accuracy or purely described requirements bring risks for project completion.

Despite significant progress in natural language understanding (NLP) and risk classification methods [1], there are problems with automation in risk identification process [2-3]. On one hand, the risk identification process is complex due to a wide variety of factors which impact the development process. On the other hand, the risk uniqueness challenge has not been resolved. It requires some form of standardisation with continuous monitoring of the best practices in risk estimation and impacts of emerging technologies.

The taxonomy of risks in software development partially resolves the risk uniqueness problem. Each concept in the taxonomy and risk identifiers are defined for each risk. Risk identifiers identify the software risks at different stages through the development phases, depending on the chosen development methodology, etc. [2] Meanwhile, when a new risk appears, we should extend the existing taxonomy. That requires the strength of subject matter experts. Without standardisation, it is difficult to resolve this problem because uniqueness is not only a unique name but also a unique meaning.

On a par with the risk uniqueness problem, we have a problem of choosing the right format for software risk representation. This problem is crucially important for risk processing automation because applied components for risk analysis might use different formats. As usual, text is the typical format for software risk representation. This format is efficient for subject matter experts (SME), ontological representation, fuzzy logic and others. Genetic algorithm (GA) uses special encoding techniques.

In the article [4], risks are presented mainly in text form and numerical tables. The authors propose a risk assessment methodology that includes determining the probability of negative events, their impact and potential losses. These indicators are presented in the form of tables with the corresponding values for each risk. While textual formats are common for risk representation, alternative methods such as GAs and fuzzy logic can be used to improve risk processing.

In [5], different applied components for risk processing were considered without a unique format or encoding specifics for risk representation. Obviously, in the era of LLM, the unique definition of software risks should use the NLP background for naming, types, methods of risk mitigation and so on. Following this approach, the embedding mechanism provides uniqueness in the meaning space of software development risks. Indeed, it is a very comfortable format for SME, for risk ordering by taxonomy, for an ontological approach, etc. Furthermore, we don't need to worry about strong naming as the uniqueness of software risks when we use meaning space. The coordinates of embedding vector define the uniqueness.

Assuming we use embedding as a basic technique for concept representation in the subject domain "Risks in Software Development". The taxonomy, which was built in [2] by SME, might be represented in vector space, but without semantics represented by the relationships in the taxonomy hierarchy. In this case, we should rely on the coordinates of embedding vectors and use the similarity between vectors to measure that semantics. By the way, the similarity doesn't prove a valid relationship between the two concepts. As usual, it is done by SMEs.

As was mentioned in [4], the authors used a tabular representation where each risk was described as a set of parameters. Indeed, it is a widespread format for risk representation because SMEs define the risks with their descriptions once and, after, continue to build the risk model using their parameters in the calculations. Referring to the risk-embedding approach, it is comfortable to use a knowledge base where each risk description is stored with an embedding vector. In the metadata, we can keep all required risk features, indicators, etc. Considering that it is impossible to describe a huge variety of possible risks related to the emerging technologies, software development methodologies and others. It is obvious that tabular representation wouldn't be efficient today. Possibly, the graph embedding provides a better representation.

Sometimes we don't need to use accurate calculations for risk estimation, some categories are enough for measurements. Fuzzy logic provides the best approach and can be included in the pipeline to automate risk processing [5-7]. Fuzzy logic involves reasoning about uncertain or imprecise information using fuzzy sets, membership functions, and inference rules. To use risk embeddings, fuzzy logic should be applied as follows [7]:

1. Define fuzzy sets for the features of the embedding risk based on the range of values for each dimension with desired thresholds.
2. Create fuzzy inference rules based on the similarity or distance between embeddings.
3. Use the fuzzy inference system to process the embeddings and generate outputs based on fuzzy rules. After, we can use a defuzzification method to convert the fuzzy output into a crisp value: the category of a sentence.

Graph embedding are more powerful on contrary of simple text embedding and can keep the semantics between concepts of subject domain, fuzzy sets with thresholds and fuzzy inference tules as well.

GAs operate through populations of candidates (chromosomes), crossover, mutation, and selection. To use embeddings in GAs, we need to structure them as chromosomes in a GA. The common GA algorithm is shown below. [8]

1. Population initialisation. Represent each risk description in the population as a vector of embeddings.

2. Fitness Evaluation. Evaluate the fitness function of each chromosome, comparing the chromosome's vector against a target using a cosine similarity.

3. Selection. Choose individuals based on their fitness.

4. Crossover. Combine two parent vectors to produce offspring. One way to do this is to take a weighted average of the parent embeddings or apply more complex operations like vector concatenation or crossing over specific parts of the embedding space.

5. Mutation. Introduce random changes to the chromosome's embedding by perturbing the vector. There are a couple of approaches here: adding small random noise to the vector, applying more sophisticated techniques like vector space transformations, shifting the embedding toward a specific semantic direction, etc.

6. Termination. Continue the process until a predefined number of generations is reached or the fitness level surpasses a certain threshold.

In case of graph embeddings, GAs can solve a wide variety of tasks. They can identify groups of risks that are more densely connected to each other, predict missing edges based on embeddings, aligning two graphs based on their embeddings, classify entire graphs or subgraphs based on learned embeddings, etc. Actually, the software development risks require graph representation to show the relationships with development phase, development stage, risk indentificators, description, risk mitigation techniques [2]. The graph embeddings significantly can help with software development risks representation for automatic processing by GAs.

Considering the proposed framework in [5], we can automate it using a graph embedding for software risk representation. The graph embedding is as a single knowledge representation format for some analytical components involved in risk estimation processes.

Based on our scientific investigation addressed to the problem of choosing the risks representation format suitable for automation, we consider a graph embedding as a valid solution. The solution offers advantages, as these embeddings provide a semantic understanding and uniqueness including relationships that text alone cannot provide. This approach relies on the similarity between graphs and vectors to represent software development risks, although SME input is required to validate relationships between concepts.

The next steps are related to the deep investigation and estimation of chosen graph embedding approach in real sotware development risk processing application.

References

1. O. G. Trofimenko, N. I. Loginova, P. O. Teslenko, O. S. Savelieva, & V. M. Polyakov (2023). Classification of Risks in Software Development Projects. Bulletin of the Kherson National Technical University, 3 (86), 119-128. doi: https://doi.org/10.35546/kntu2078-4481.2023.3.15.

2. M.Y. Lyashkevych, I.Y. Rohatskiy, V.Y. Lyashkevych, R.Y. Shuvar (2024). Software risk taxonomy creation based on the comprehensive development process.

Electronics and Information Technologies, 1(27), 59-71. doi: https://doi.org/10.30970/eli.27.5.
3. G. Crispin (2020). The Essence of Risk Identification in Project Risk Management: An Overview. International Journal of Science and Research (IJSR), 9, 1553-1557. doi: https://doi.org/10.21275/SR20215023033.
4. Hrytsiuk, Yu. I., & Zhabych, M. R. (2018). Risk management of the implementation of program projects. Scientific Bulletin of UNFU, 28(1), 150–162. doi: https://doi.org/10.15421/40280130.
5. M. Lyashkevych (2025). A hybrid intelligent framework for the accuracy improvement of the collection and completeness of software requirements formalisation. Proceedings of the 2nd International Scientific and Practical Internet Conference "Transcending Boundaries: Unravelling the Dynamics of Cutting-Edge Research and its Transformative Impact on the Global Sphere", Dnipro, Ukraine, July 11-12, 2025 - Dnipro, 85-87. Url: http://www.wayscience.com/wp-content/uploads/2025/07/Conference-Proceedings-July-10-11-2025-1.pdf.
6. James, Stephy & Renjith, V.R. (2024). A Comprehensive Review on the Application of Fuzzy Logic in Risk Analysis. International Journal of Computer Applications, 186, 16-38. doi: https://doi.org/10.5120/ijca2024924254.
7. Atlam, H. F., Walters, R. J., Wills, G. B., & Alhazmi, O. H. (2021). Fuzzy logic with expert judgment to implement an adaptive risk-based access control model for IoT. Mobile Networks and Applications, 26(5), 2545–2557. doi: https://doi.org/10.1007/s11036-019-01214-w
8. Bouabdallaoui, I., Guerouate, F., & Sbihi, M. (2024). Hybrid Text Embedding and Evolutionary Algorithm Approach for Topic Clustering in Online Discussion Forums. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, 13(1), e31448. doi: https://doi.org/10.14201/adcaij.31448

МОДЕЛЮВАННЯ ПЕРЕРОЗПОДІЛУ КОМПОНЕНТІВ ПЕРЕД ФРОНТОМ ПЕРЕТВОРЕННЯ ФЕРИТ → АУСТЕНІТ + КАРБІД ЗА УМОВ ДИФУЗІЙНОЇ ЗМІНИ СКЛАДУ

Чорноіваненко Катерина Олександрівна
кандидат технічних наук, доцент
Повзло Євген Леонідович
Кафедра «Систем якості, стандартизації та метрології»
Український державний університет науки і технологій

У даній роботі досліджено концентраційні поля перед фронтом при трифазному перетворенні $\alpha \rightarrow \gamma$+карбід під час навуглецювання. Об'єктом дослідження слугували феритні сплави заліза, леговані двома елементами Me1 і Me2. Знаючи розподіл концентрацій перед фронтом перетворення, можна прогнозувати його стійкість методом збурень [1]. Аналіз росту евтектоїдних