Tree Visualization of Patient Information for Explainability of AI Outputs

Sandeep Ramachandra[1,*,†], David Vander Mijnsbrugge[1,†], Pieter-Jan Lammertyn[2], Stijn Dupulthys[2], Femke Ongenae[1] and Sofie Van Hoecke[1]

[1]*IDLab, Ghent University-imec, Ghent, Belgium*
[2]*Radar, AZ Delta, Roeselare, Belgium*

Abstract
Knowledge graphs (KG) and ontologies can be leveraged to efficiently convey information and provide a great aid in explaining the outcomes of neural networks in the healthcare domain. In this short research paper, we introduce a novel approach that encodes patient information and expert knowledge of diseases into a single temporal graph which enables seamless integration into neural networks. Furthermore, we present a visualization tool that explains the output generated by these networks, leading to better understanding of the provided decisions by healthcare professionals and other stakeholders.

Keywords
Explainable AI, Visualizations, Dynamic graphs, Temporal graphs

1. Introduction

Patient health records are multi-modal in nature, containing a combination of structured data (e.g. performed procedures and made diagnoses), free text (e.g. doctor notes), and higher dimensional data, such as images (e.g. X-ray and CAT scans) or time series (e.g. heart beat measurements). Moreover, patient information is dynamic in nature and constantly evolves over time, e.g. changing laboratory test results over subsequent hospital visits, and/or the condition of the patient changing over time. This makes patient information an inherently complex data source to work with.

To enable proper analysis, e.g. with Machine Learning (ML), it is important to be able to represent this multi-modal data in a data structure which makes the correlations between the data explicit. Due to their expressive power, Knowledge Graphs (KGs) and ontologies have become increasingly popular to encode such patient information [1]. Moreover, their graph representation allows visual and easily understandable interpretation of the information. An ontology enables us to additionally encode the established expert-knowledge about the

The International Semantic Web Conference 2023
*Corresponding author.
†These authors contributed equally.
sandeep.ramachandra@ugent.be (S. Ramachandra); david.vandermijnsbrugge@ugent.be (D. V. Mijnsbrugge); pieter-jan.lammertyn@azdelta.be (P. Lammertyn); stijn.dupulthys@azdelta.be (S. Dupulthys); femke.ongenae@ugent.be (F. Ongenae); sofie.vanhoecke.ugent.be (S. V. Hoecke)
0000-0001-5505-4362 (S. Ramachandra)

healthcare domain and connect it to the data in the KG, allowing for the incorporation of correlations between symptoms and diseases in the KG. Leveraging the incorporated prior knowledge empowers more effective data analytics and enhances performance. On the other hand, the utilization of timing information in KGs, typically represented using date time nodes attached within the same graph, proves less optimal when analyzing patient information, as the timing information represents a change in the graphs itself. In order to facilitate this interpretation, the patient information can be represented as graphs that dynamically change over time rather than using time nodes in the complete graph.

At the same time, there has been a surge in the popularity of machine learning (ML) methods and graph embedding techniques that empower the execution of advanced data analytics using these KGs. Prominent examples include Graph Convolutional Networks (GCNs) and RDF2Vec [2]. In critical domain like healthcare, eXplainable AI (XAI) is gaining increasing importance [3]. As any erroneous output can be harmful to the patient, the healthcare expert, and potentially other non-AI experts (e.g. care provider, patient), should be able to clearly trace the significant input features contributing to the output of the ML. Furthermore, it is essential that this explanation aligns with, or is substantiated by, the expert knowledge of the domain, which is captured in the ontology. While some recent graph embedding methods try to retain the interpretable aspects of KGs, e.g. INK [4], most graph embedding methods are considered black-box, especially the ones based on deep learning techniques, such as GCNs. The latter are therefore often combined with post-hoc explanation methods aiming to elucidate the model's output for specific inputs, such as SHAP[5] and Saliency Maps[6]. To do so, these methods offer insights into the contribution of each input feature towards generating the final output, and thus into an importance factor of each feature. This feature contribution can also be used in KGs to visualize the importance factor of each node in the KG using a color scale over the post-hoc explanation of the output of the AI for any given task. Prior works on visualization and post-hoc explanations for KGs do not take into account the dynamic temporal patient data [1].

Therefore, in this short research paper, we propose a novel methodology to encode dynamic temporal patient information in a KG and to visualize the output of the graph embedding model in a clear and understandable manner by visualizing the real world patient information along the time axis as well as visualizing the importance factors for each entity in the patient graphs for ease of comparison for a healthcare expert.

2. Related work

There are two domains coming together in this paper, namely patient information encoding in a KG optimized for data analytics, and graph visualization optimized for exploration by domain experts interpreting the output of these analytics. We therefore explore the state-of-the-art in these two domains below.

Graph encoding of patients With the design of ontologies, such as SNOMED [7] and OMOP [8], a lot of work has already been performed on representing patient information in KGs. However, most of these representations ignore the temporal dimension of patient data and fail to fully harness the intrinsic explainability offered by graph representations. In Choi et al. [9], the authors recognised the EHR data as being multilevel with diagnosis being related to certain treatments and utilise this property to improve the performance of their model. However, most concepts are interrelated to each other and lack fixed hierarchical levels. This shortcoming is solved by [10] with the representation of patient visits being in graph structures. In [11], the authors combine a patient graph with ontologies to improve the quality of the embeddings of the concepts. These papers show the potential of graph neural networks with patient data but fail to explain their output which is critical in the healthcare domain.

Graph visualization There is no one size fits all solution for graph visualization [12]. Every use case for ontology and KG visualization utilises different tools and focuses on different aspects of visualization. For example, VOWLExplain [13] utilised WebVOWL [14], a web based ontology visualization tool, to visualize patient information from The Cancer Genome Atlas and visually explain the recommendations of an AI model. Their user study showed that the graph explanations of AI recommendations regarding the patient were equally accurate and comprehensible compared to textual explanations. While the paper showcases the explainability inherent to KGs, they did not explore the temporal aspect of patient information. In [15], the Neo4j graph database is explored for a healthcare case where they showcase the patient's progression along the time axis. The presentation was largely exploratory and did not use the explainability offered by graphs. So to conclude, state-of-the-art focuses on either the temporal KGs or on explainability, where we are the first to tackle the combination of both.

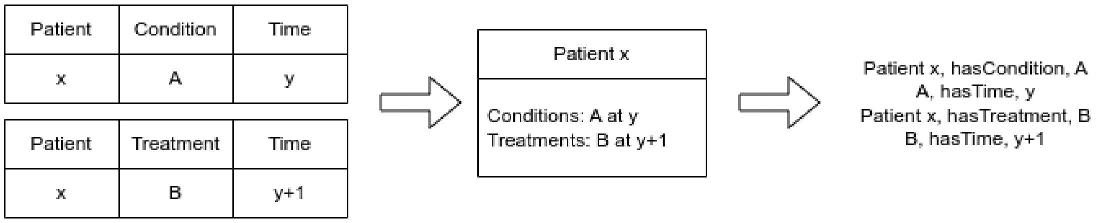
3. Methodology for representing and visualizing patient information in a KG optimized for explainable AI

In this section, we first highlight the requirements for the proposed KG representation method to enable XAI with Graph Neural Networks (GNN), and then dive deeper into the proposed methodology itself and the accompanying visualization.

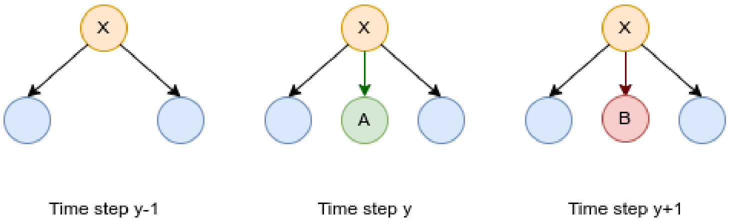
3.1. Requirements

The requirements for the encoding of a patient graph are:

- **Dynamic Temporal view**: The patient graph should be separable into different time steps since patient data is added *incrementally* over time, e.g. per visit to the hospital. Real-world patient data is constantly updated according to the changing condition of the patient and performed procedures, e.g. new laboratory results, change in medication, and new diagnoses. The visualization should reflect this. Furthermore, this incremental nature should result in a much less cluttered representation of the data, especially after prolonged periods of time.
- **Tree view**: Ideally, the patient data is structured in a tree-based manner as this accurately reflects the observatory nature of the data. Namely, each specific diagnosis, treatment, observation, laboratory observation, etc. should be *unique* in the representation, as they are likely to be *repeated* across visits, e.g. chemotherapy is a treatment that is repeated many times over a given period where each instance of the treatment carries relevant information. The patient itself represents the trunk of the tree. Each visit then represents a branch in the tree, with all the data connected to that visit, e.g. observations, treatments,



(a) Triples generation



(b) Time separation

Figure 1: Toy example illustrating steps 1 and 2 of the methodology. Subfigure 1a shows a triple generation for data fragment of one patient. Subfigure 1b shows the addition of the data to graphs at the time step y and y+1, with graph at time step y-1 showing patient information, e.g. gender which is not varying with time.

and diagnoses made, as subbranches within that branch resulting in a hierarchical tree structure with variable length branches.

- **Ontology**: It is important to be able to link the data in the KG to the prior knowledge encoded in the ontology, as this delivers important information to the XAI. For example, each diagnosis and treatment in the procedure are linked to each other in the ontology.

3.2. Methodology

Our novel method to generate a patient graph containing the needed information meeting the requirements set out in Section 3.1 consists of the following 4 steps:

1. **Triple generation**: As a first step, the multi-modal data has to be collected from their original representation, e.g. a relational database or data lake, into an intermediary representation so that one patient's data (Conditions, Treatments, Drugs, Labs, Observations) is encapsulated in one object along with their associated timestamp. Each concept node has to be made unique so that the nodes do not get conflated in the graph (See *Tree* requirement in Section 3.1). As ontology, the OMOP (Observational Medical Outcomes Partnership) [16] ontology is used. Using relationships set out in this OMOP schema, information on each object is converted into a N-Triples (NT) file that is used to generate the patient graph. Figure 1a shows a toy example for this triple generation.
2. **Time separation**: Next, each timestamped concept present in the generated NT file has to be separated along the time axis. To do so, each time step can be defined as either the time when a discrete change in patient data occurs, or as a fixed change in time. For this paper, we chose discrete changes as this way the changes are immediately apparent.

Discrete changes are also better when dealing with patient data that extends over years, for example for chronic (long term) diseases, in the NT file for each patient, we then create a new NT file for the new graph per time step. A quad file can also be created with each time step as a disjointed graph as this would behave in a similar manner. Figure 1b continues with the toy example showing a visualization of the graph separated on the time dimension.

3. **Adding ontology**: The ontology of OMOP is contained in two tables in OMOP schema, i.e. Concept Ancestor and Concept relationship, with both incoming and outgoing relations defined. To comply with the *Tree* requirement (see Section 3.1), only the outgoing relations are taken. In this step, from each patient's NT file, the leaf nodes are used to look for matching subject nodes in two tables. The filtered triples are connected to each leaf node in patient graph, duplicated across the time axis.
4. **Visualization**: As the visualizations have to be dynamic and compatible with the frameworks of the neural networks, we chose Plotly [17] due to its interactive features and ease of use to enable the time axis visualization in the fourth and final step. To do so, each time step of the patient graph is added as a trace to the Plotly figure. Since the patient graph is modelled as a tree, we use the Reingold-Tilford layout to spread out the nodes and layer of the tree for better visibility. Additional information about each node is added to a hover interaction with the node. For visualizing the explanation, the contribution of each node to the output of the neural network can be calculated/inferred. For example, attention blocks [18] can be used to gather this information. As such, an importance is calculated for every node which can be assigned as a color to each node, allowing for instant visibility of the importance of each node at each time step. One can also highlight the time step with the most contribution by surfacing it first while viewing the visualization. This can be interesting, for example, for healthcare professionals.

4. Use case: Lung cancer patient representation and visualization

We applied the presented method on a synthetic lung cancer patient, designed together with clinical experts from AZ Delta, to empirically prove applicability of our method.

Figure 2 shows a screenshot of an interactive example patient visualization developed using the methodology presented in Section 3.2. Due to the example being synthetic patient data, the importance factors are assigned randomly. A live example can be found at https://predict-idlab.github.io/Tree-Visualization/. The example shows an overview of the patient information as well as the sub-graphs for related concepts, extracted from the OMOP ontology and attached to each instance of the concepts. The example visualizes the patient KGs with the temporal information retained and showcases the explainability possibilities of the patient KGs. The visualization clearly displays the most important nodes in a darker shade of red so that a healthcare professional can, without any deep understanding of the model used, immediately notice the most significant node to the model at that time step. The same conclusion can be made across the time dimension as the time step that is the most significant, can be highlighted by displaying it by default.

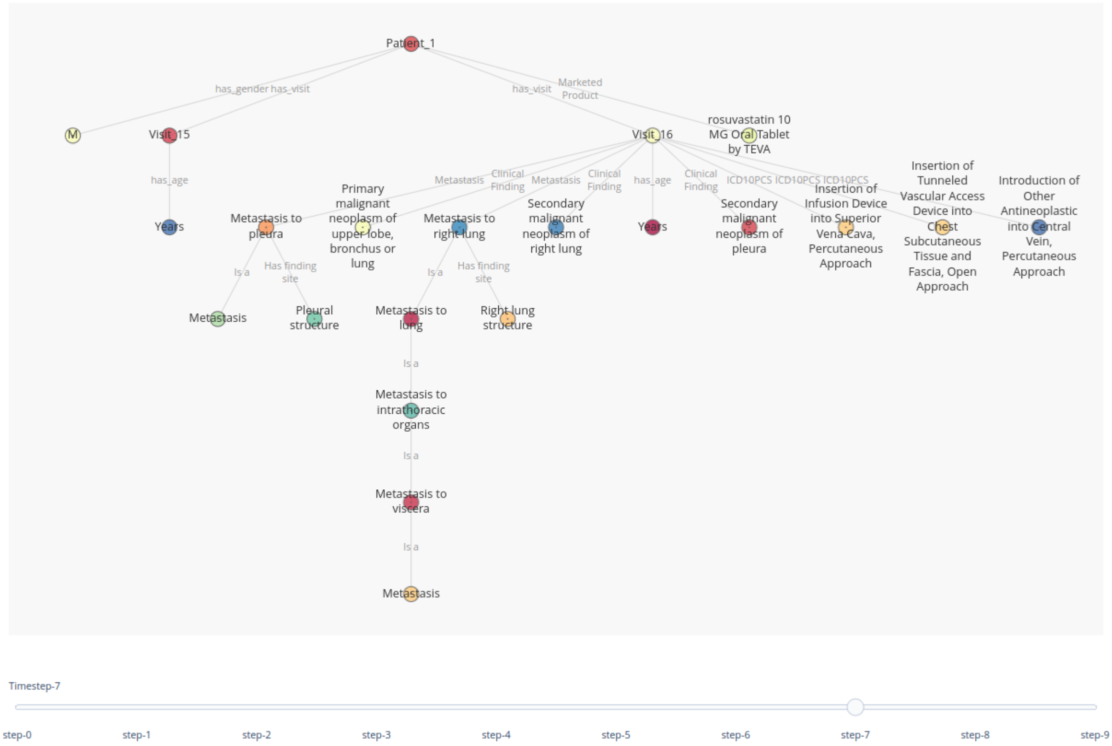


Figure 2: Screenshot of the visualization of the (synthetic) patient data. The colors indicate the importance factor for each node for the output of the AI. The importance factor used here are assigned randomly. The live example can be found at https://predict-idlab.github.io/Tree-Visualization/.

5. Conclusion

We have proposed a novel methodology for the representation and visualization of patient information from multi-modal patient information while also incorporating the time dimension. Bringing this time dimension to patient representation and visualization is important to enable us to utilise the dynamic nature of the patient data. The resulting representation can be used in a Graph Neural Network (GNN) for various downstream tasks. Our method explains the output of this GNN using importance factors which are incorporated in the visualization as node coloring. This provides a visually easily interpretable explanation of the GNN output. We showcased the correct functioning of our methodology on a lung cancer use case, using synthetic patient data. In the near future, we hope to add interactive views of the graph with highlighting paths for important nodes and real time visualization of graphs.

Acknowledgments

This study was partially funded by the Flanders AI Research Program and the VLAIO O&O ADAM project with AZ Delta (HBC.2020.3234).

References

[1] J. Schrodt, A. Dudchenko, P. Knaup-Gregori, M. Ganzinger, Graph-Representation of Patient Data: a Systematic Literature Review, J. Med. Syst. 44 (2020) 86–7.

[2] P. Ristoski, H. Paulheim, Rdf2vec: Rdf graph embeddings for data mining, in: ISWC 2016, 2016, p. 498–514.

[3] A. Holzinger, C. Biemann, C. S. Pattichis, D. B. Kell, What do we need to build explainable ai systems for the medical domain?, 2017. arXiv:1712.09923.

[4] B. Steenwinckel, G. Vandewiele, M. Weyns, T. Agozzino, F. D. Turck, F. Ongenae, Ink: knowledge graph embeddings for node classification, Data Mining and Knowledge Discovery 36 (2022) 620–667.

[5] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017).

[6] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, arXiv preprint arXiv:1312.6034 (2013).

[7] W. S. Campbell, J. Pedersen, J. C. McClay, P. Rao, D. Bastola, J. R. Campbell, An alternative database approach for management of snomed ct and improved patient data queries, Journal of Biomedical Informatics 57 (2015) 350–357.

[8] B. Aldughayfiq, F. Ashfaq, N. Z. Jhanjhi, M. Humayun, Capturing semantic relationships in electronic health records using knowledge graphs: An implementation using mimic iii dataset and graphdb, Healthcare 11 (2023).

[9] E. Choi, C. Xiao, W. F. Stewart, J. Sun, Mime: Multilevel medical embedding of electronic health records for predictive healthcare, 2018. arXiv:1810.09593.

[10] E. Choi, Z. Xu, Y. Li, M. W. Dusenberry, G. Flores, Y. Xue, A. M. Dai, Graph convolutional transformer: Learning the graphical structure of electronic health records, 2019. arXiv:1906.04716.

[11] Z. Lin, D. Yang, X. Yin, Patient similarity via joint embeddings of medical knowledge graph and medical entity descriptions, IEEE Access 8 (2020) 156663–156676.

[12] M. Dudáš, S. Lohmann, V. Svátek, D. Pavlov, Ontology visualization methods and tools: a survey of the state of the art, The Knowledge Engineering Review 33 (2018).

[13] F. Serrano, S. Nunes, C. Pesquita, Vowlexplain: Knowledge graph visualization for explainable artificial intelligence, VOILA - ISWC (2022).

[14] S. Lohmann, S. Negru, F. Haag, T. Ertl, Visualizing ontologies with vowl, Semantic Web 7 (2016) 399–419.

[15] J. Roemer, Improving patient outcomes with graph algorithms, 2020. URL: https://neo4j.com/blog/improving-patient-outcomes-algorithms-graphconnect.

[16] OMOP CDM v5.4, 2023. URL: https://ohdsi.github.io/CommonDataModel/cdm54.html.

[17] Plotly Technologies Inc., Collaborative data science, 2015. URL: https://plot.ly.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017. arXiv:1706.03762.