On Graph Construction for Classification of Clinical Trials Protocols Using Graph Neural Networks

Ferdowsi, Sohrab; Copara Zea, Jenny Linet; Gouareb, Racha; Borissov, Nikolay; Jaume-Santero, Fernando; Amini, Poorya; Teodoro, Douglas

How to cite

FERDOWSI, Sohrab et al. On Graph Construction for Classification of Clinical Trials Protocols Using Graph Neural Networks. In: International Conference on Artificial Intelligence in Medicine. AIME 2022. Cham : Springer, 2022. p. 249–259. (Lecture Notes in Computer Science) doi: 10.1007/978-3-031-09342-5_24

This publication URL: https://archive-ouverte.unige.ch//unige:162732

Publication DOI: 10.1007/978-3-031-09342-5_24

On graph construction for classification of clinical trials protocols using Graph Neural Networks

Sohrab Ferdowsi$^{1,2[0000-0003-3768-6408]}$, Jenny Copara$^{1,3[0000-0002-1510-3331]}$, Racha Gouareb$^{1[0000-0001-6611-2548]}$, Nikolay Borissov$^{4[0000-0002-8423-9873]}$, Fernando Jaume-Santero$^{1,2[0000-0002-8441-3798]}$, Poorya Amini$^{4[0000-0002-9473-0172]}$, and Douglas Teodoro$^{1,2,3[0000-0001-6238-4503]}$

[1] Department of Radiology and Medical Informatics, University of Geneva, Geneva, Switzerland
[2] Business Information Systems, University of Applied Sciences and Arts of Western Switzerland (HES-SO), Geneva, Switzerland
[3] Swiss Institute of Bioinformatics, Lausanne, Switzerland
[4] Risklick AG, Bern, Switzerland

Abstract. A recent trend in health-related machine learning proposes the use of Graph Neural Networks (GNN's) to model biomedical data. This is justified due to the complexity of healthcare data and the modelling power of graph abstractions. Thus, GNN's emerge as the natural choice to learn from increasing amounts of healthcare data. While formulating the problem, however, there are usually multiple design choices and decisions that can affect the final performance. In this work, we focus on Clinical Trial (CT) protocols consisting of hierarchical documents, containing free text as well as medical codes and terms, and design a classifier to predict each CT protocol termination risk as "low" or "high". We show that while using GNN's to solve this classification task is very successful, the way the graph is constructed is also of importance and one can benefit from making a priori useful information more explicit. While a natural choice is to consider each CT protocol as an independent graph and pose the problem as a graph classification, consistent performance improvements can be achieved by considering them as super-nodes in one unified graph and connecting them according to some metadata, like similar medical condition or intervention, and finally approaching the problem as a node classification task rather than graph classification. We validate this hypothesis experimentally on a large-scale manually labeled CT database. This provides useful insights on the flexibility of graph-based modeling for machine learning in the healthcare domain.

Keywords: Graph Neural Networks · Machine Learning · Natural Language Processing · Clinical Trials · Healthcare Informatics

1 Introduction

Healthcare-related events and the underlying clinical data sources are typically highly heterogeneous, irregular, consisting of multiple modalities and dealing with various semantic representations [13], [24]. The patients records during multiple visits to care centers, the large body of medical text generated in hospitals, the multiple imaging modalities required for diagnosis and various other sources like lab reports are potentially all relevant in healthcare practice [22]. A natural choice to model these variations in a unified manner would be the use of graphs, where nodes, edges and features have the flexibility, as well as the capacity to hold these interrelated sources of data, and under many different scenarios [23], [22].

While the literature of machine learning and its related fields has evolved primarily to deal with regular grid-like sources of data, recent years have seen significant activities to generalize machine learning concepts to graph-based data. This has given rise to the field of geometric deep learning [1] with high promise and noticeable success using Graph Neural Networks (GNN's) across various disciplines that can benefit from graph-based representations (see e.g., [10] and [21] on the use of GNN's in natural sciences).

The domains of healthcare informatics and machine learning in medicine, therefore, have seen significant activities in this direction and many works have shown promising results in the integration of GNN's to tackle healthcare-related problems. As an example, the work of [3] uses GNN's to supplement Electronic Health Records (EHR) with hierarchical information, showing noticeable improvements in diagnosis prediction compared to Recurrent Neural Networks (RNN's). Similarly, the work of [20] uses GNN's combined with neural language models to better capture the hierarchical structure of medical codes and perform medication recommendation from EHR's. Several prediction tasks again based on EHR's are addressed in the work of [31], where the authors propose a regularization technique to improve the robustness of training.

While the use of GNN's has been shown to be very effective, in this work, we show that the way the graph is constructed and the problem is formulated is also of prime importance. In particular, we build up on our prior work [7], where we encode Clinical Trials (CT) protocols in a hierarchical graph to predict their termination risk, by further linking them with edges according to connections between the CT phases, conditions and interventions. Therefore, rather than considering multiple disconnected graphs and posing the problem as graph classification, we consider a single but very large graph and target a node classification problem to classify CT protocols. Interestingly, while all these newly considered edges between graphs arise directly from the CT protocols and do not contain any extra information that is not already encoded as node features, we show that under all setups, this new formulation, not drastically but very consistently improves the CT classification performance.

2 Background and related works

In this section, we showcase the required backgrounds and review some of the relevant methods and efforts in the literature. Section 2.1 discussed data-driven CT risk analysis, which is the main task we target in this paper. Since our proposed methodology is the use of GNN's to tackle this problem, we review some basic concepts of graph-based machine learning in section 2.2. We then very briefly discuss the concept of text featurization in section 2.3.

2.1 Data-driven efforts for CT risk analyses

The systematic way to assess the safety and efficacy of candidate clinical interventions and medications for the treatment of medical conditions is to carry out randomized studies, a.k.a. Clinical Trials (CT's), on volunteer subjects and during multiple phases. Because of their complexity and extent of the resources needed for these studies, they take around 60-70% of the average 13.8 years long drug development cycle [18] and constitute a major portion of the average estimated 1.3B$ cost for drug development [29]. In spite of the very strict guidelines in place from healthcare authorities and careful planning of trials prior to their execution, unfortunately, no more than only 14% of CT's manage to continue from phase 1 to the market approval [28]. Therefore, in order to minimize these costs and the associated risks, it would be highly beneficial to try to optimize CT protocols prior to their implementation.

Although there are various government registries that provide access to past and current CT records to the public, there has been only few works in the literature reporting data-driven methods to assess the behavior of CT's based on simple risk measures. The works of [9] and [11] use traditional data-mining techniques to classify termination risk of CT's. The more recent work of [5] uses hand-crafted features and feeds them to off-the-shelf classifiers to target "completed" and "terminated" CT status categories. Similar methodologies have been developed in [6] to assess COVID-19 CT's.

To benefit from the power of end-to-end deep learning and to avoid using hand-crafted approaches, our recent work [7] targeted the CT classification problem using GNN's. This was motivated by the highly hierarchical structure of CT's as shown in Fig. 1, where besides their textual content, also the structure of the protocols was shown to be relevant for risk classification, an assumption that was strongly corroborated by the significant performance gains reported. In this work, we revisit this approach by reformulating the graph-based CT classification and show performance improvements. Before presenting our proposed method in section 3, we briefly review some basic concepts from graphs and graph-based machine learning.

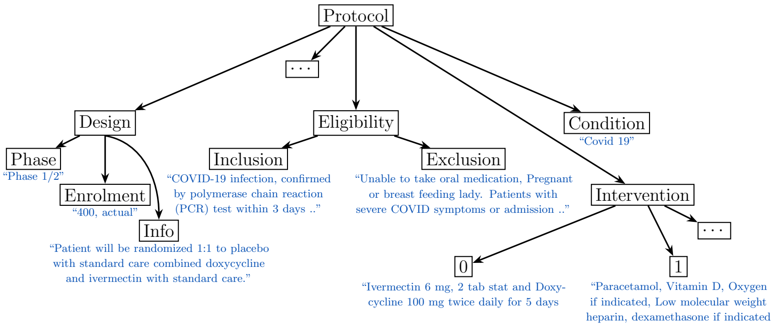
Protocol
· · ·
Design
Eligibility
Condition
"Covid 19"
Phase
"Phase 1/2"
Enrolment
"400, actual"
Info
"Patient will be randomized 1:1 to placebo with standard care combined doxycycline and ivermectin with standard care."
Inclusion
"COVID-19 infection, confirmed by polymerase chain reaction (PCR) test within 3 days .."
Exclusion
"Unable to take oral medication, Pregnant or breast feeding lady. Patients with severe COVID symptoms or admission .."
Intervention
· · ·
0
"Ivermectin 6 mg, 2 tab stat and Doxycycline 100 mg twice daily for 5 days
1
"Paracetamol, Vitamin D, Oxygen if indicated, Low molecular weight heparin, dexamethasone if indicated

Fig. 1: Simplified schematic view of a CT protocol. Leaf nodes contain free text and medical codes. While the top parent nodes are fixed, children nodes have variable structure across CT examples within the same registry.

2.2 Graph neural networks

In its most abstract, yet practically powerful form for many applications, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}; \mathcal{X})$ consists of node sets $\mathcal{V} = \{v_1, \cdots, v_{|\mathcal{V}|}\}$, the set of edges $\mathcal{E}$ with pairs of nodes (u_i, v_i), which denote the existence of an edge between the two nodes $u_i, v_i \in \mathcal{V}$, as well as a set of features $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_{|\mathcal{V}|}\}$ associated to each of the nodes (and/or also to the edges in some applications).

Graph Neural Networks (GNN's) try to generalize the deep learning practice and machine learning concepts to a graph $\mathcal{G}$, or multiple graph objects $\{\mathcal{G}_1, \cdots, \mathcal{G}_N\}$. This, however, is more challenging to deal with than the case of regular grids like images, text, sound or time series data. The difficulty lies primarily in that, unlike grids, there does not usually exist a canonical way to order nodes of a given graph. Hence, all machine learning steps should be agnostic to node orderings[5], both within a graph and also across multiple graphs.

A largely successful approach to tackle this permutation ambiguity is the Message Passing (MP) paradigm of [12], which replaces some of the usual list operation steps of machine learning with set operations that are order-agnostic. Concretely, for a node $u \in \mathcal{V}$ the nodes $v \in \mathcal{N}(u) = \{v \in \mathcal{V} | (v, u) \in \mathcal{E}\}$ in its immediate neighborhood send a "message" using a generic differentiable "aggregation" operation $\mathbb{A}\{\cdots\}$ on their features. This is then used to "update" the features of $\mathbf{u}$, using another generic differentiable operation $\mathbb{U}[\cdot, \cdot]$. These steps can be summarized as:

$$\mathbf{x}_u^{[l+1]} = \mathbb{U}\left[\mathbf{x}_u^{[l]}; \mathbb{A}\left\{\mathbf{x}_v^{[l]}, \forall v \in \mathcal{N}(u)\right\}\right], \tag{1}$$

where super-scripts $1, \cdots, l, \cdots, L$ refer to the fact that this operation is carried out L times, and starting from initial raw features $\mathbf{x}^{[1]} = \mathbf{x} \in \mathcal{X}$. After the L iterations of MP, each $\mathbf{x}_v^{[L]}, v \in \mathcal{V}$ has aggregated features from its L-hop neighbors, so that the content from both the initial raw features, as well as the topology of the graph are captured within the final features. Famous instances of these generic operations are the Graph Convolutional Networks (GCN) from [17], the Graph Attention Network (GAT) from [26], or the GraphSAGE operator of [16], among many others.

Certain machine learning tasks on graphs, e.g., node classification, node regression or link prediction, are performed locally and on the node level, rather than globally on the whole graph. For these tasks, the resultant feature of every node, i.e., $\mathbf{x}_v^{[L]}, v \in \mathcal{V}$, or perhaps a subset of nodes (like those from train, validation or test splits) can further follow processing steps like typical Multi Layer Perceptron (MLP) (i.e., multiple affine layers with non-linearities in between) to be matched against some label information. A common scenario in these cases is that the nodes from all data splits ($\mathcal{V}_{\text{train}}$, $\mathcal{V}_{\text{valid}}$ and $\mathcal{V}_{\text{test}}$) are present at the time of training. However, only the feature information of the test set is used during MP operations and their label information is of course not used during training and loss calculation. This scenario is referred to as the "transductive" case[6], as opposed to the "inductive" case where the test nodes are entirely absent during training.

[5] More technically, they should be either "permutation invariant" or "permutation equivariant" to the order of nodes.

[6] which resembles semi-supervised classification in some sense

On the other hand, other machine learning tasks on graphs, like graph classification, regression or generation, are performed globally and on the whole graph. For these tasks, before continuing from the features $\mathbf{x}_v^{[L]}, v \in \mathcal{V}$ to the target labels, a global "pooling" stage $\mathbb{P}_G\{\cdots\}$ is needed to provide a global representation $\mathbf{z}_{\mathcal{G}_j}$ for the whole graph from individual nodes. While this should again be an order-agnostic set operation, a simple averaging of features is usually sufficient at this stage, since the MP algorithm has already integrated the topological content to the features. For each of the graphs in a given split $\{\mathcal{G}_1, \mathcal{G}_2, \cdots\}$, the pooled representations and their corresponding labels $\{(\mathbf{z}_{\mathcal{G}_1}, y_1), (\mathbf{z}_{\mathcal{G}_1}, y_1), \cdots$ are then treated as typical machine learning feature-target pairs and can be fed to MLP's with standard training recipes.

Note that for both these cases, i.e., the node-level and graph-level scenarios, since the whole pipeline is designed to be differentiable, end-to-end training using stochastic gradient descent is possible, as is the case for other deep learning tasks. However, the concept of mini-batching, while very straightforward in grid-like data, is more intricate for graph-based data and in particular the node-level tasks, since the connectivity of the nodes should somehow be taken into account during random sampling of the nodes. Examples of approaches to tackle this issue are the works of [2] and [30].

2.3 Text featurization

A crucial step in doing machine learning on text is to perform text featurization to come up with unified-length vectors as representations of textual content. Since vectorial representations are very fundamental for machine learning algorithms, this basic step has been extensively studied within NLP communities and various generations of methods have been proposed. Among the earliest efforts in this direction is the use of Bag-Of-Words (BOW), where the frequency of the appearance of the tokenized items of the collection within a piece of text is considered as its vectorial features. While an important difficulty to do machine learning on such representations is the high dimensionality imposed by the number of the tokens of the collection, one can benefit from their high sparsity to project them to much lower dimensions, as e.g. in [7], where a very practical setup has been implemented suitable for the classification task and with low latencies.

The fundamental shortcoming with BOW-based representations, however, is that they disregard entirely the token context within the text sequence. The state-of-the-art approach to account for this sequential structure is the transformers of [25], where relying on the (self-) attention mechanism, they achieve significant improvements across many tasks, as e.g., in [4]. Due to their very demanding computational complexities, however, instead of always considering a large transformer model within the typical end-to-end machine learning loop, an active line of work (e.g., as in [19]) tries to benefit from them to embed text to vectors, while freezing the transformer weights and obviating the need to always backpropagate the errors through them.

3 Proposed framework

We now described different elements used in our framework.

3.1 Graph formation

Individual CT's: A typical biomedical text, notably our example of CT protocols, is usually structured in a hierarchy of different components. This hierarchy can be translated to graphs, most commonly as trees, similar to the example of Fig. 1.

In our case, for every individual CT protocol, pieces of text appear as leaf-nodes, where they are featurized to fixed-length vectors. As a simple baseline, here we use the BOW-based featurization described in section 2.3, which is very fast to execute. To increase performance of featurization, we also use the contextual text embedding approach using transformers which benefits from pre-training on medical text.

These vectorial features will then constitute $\mathbf{x}_v^{[0]} \in \mathcal{V}$, in our terminology of section 2.2, while non-leaf nodes without content will be initialized with all-zero vectors of the same dimension ($d = 768$). This is depicted in Fig. 2 (top), and is performed for all individual CT's of the collection.

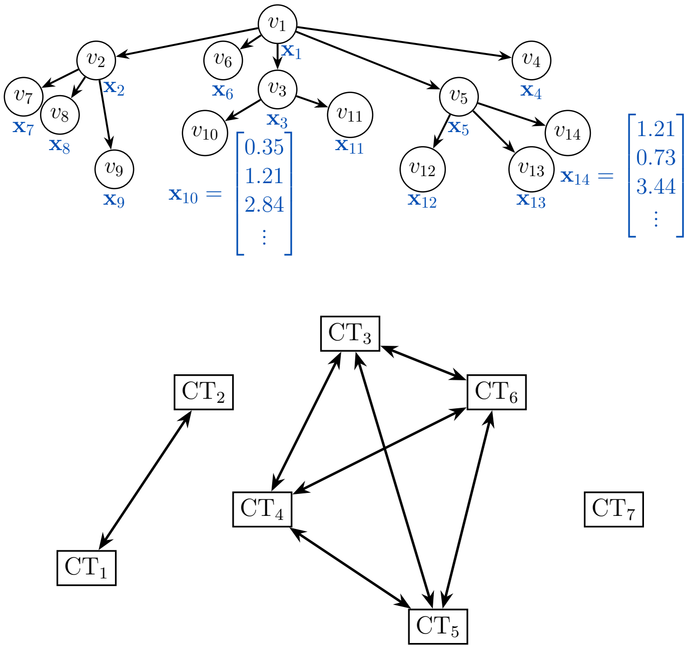

Fig. 2: (top) A sample CT document in the collection forming a graph with nodes consisting of featurized text. (bottom) Connecting graph of the documents in the collection as super-nodes. Each CT document contains a similar sub-graph as in the top figure.

All CT's as one graph: The above setup considers each protocol independently, so a graph-classification task can be performed to classify them, as in our prior work [7]. As we propose in this work, however, certain criteria can be used to connect these individual CT protocols together so that the MP algorithm of equation 1 can benefit from richer and more meaningful connections between the CT's. This implies a single but very large graph containing the individual protocols as super-nodes, keeping all their nodes, edges and features within the large graph. As sketched in Fig. 2 (bottom), similar CT's can send messages between one another during the steps of the MP algorithm.

As for the criteria to connect the CT protocols together, we consider the fields "condition", "intervention", and "phase" of the CT's as important aspects where CT's can be considered as similar. In one model abstraction, we connect every CT, i.e., every super-node of the large graph, if they have at least one condition (among possibly many) and one intervention in common. In the second abstraction, we further require them to have the same trial phase. To benchmark this idea against the case where CT's are considered entirely independently (e.g., as in [7]), we consider these super-nodes as an empty graph, i.e., without any edges between them, while of course considering all the nodes and edges within the super-nodes. Table 1 provides some basic statistics of the connecting graph induced by these 3 cases.

connecting graph	# nodes	# edges	avg. degree	# connected components
empty	164326	0	0	164326
cnd.+int.	164326	4766646	3.53e-04	95033
cnd.+int.+ph.	164326	1896404	1.40e-04	102633

Table 1: Basic statistics of the connecting graphs (considering only super-nodes)

3.2 Classification

Once the large graph corresponding to all CT protocols in the collection is formed, we perform $L = 5$ stages of the MP algorithm of equation 1. Note, however that this graph may not entirely fit within a GPU, as e.g., in our case we have more than 15.4 million nodes overall (an average of around 94 nodes per each super-node), as well as more than 24.7 million edges. We therefore need to sample nodes and edges prior to training. In order not to lose the correspondence of nodes and edges, random sampling should be avoided and more meaningful sampling strategies that keep connected nodes within the same bag should be preferred. In our case, we use the cluster-GCN algorithm [2] to sample connected nodes by benefiting from graph clustering techniques.

Finally, once the MP algorithm is run on the sub-graphs and the raw features of all nodes are enriched with connectivity information, all the nodes belonging to the same CT protocol are pooled together to provide a final representation of the CT. This is then followed by an MLP to provide the final class outputs with softmax, where they are matched against the target labels using a class-weighted cross-entropy loss.

4 Experimental Results[7]

Data preparation: We use the publicly available *ClinicalTrials.gov*[8] with more than 360K CT items Similar to the setting described in [7], we exclude the CT's whose status is not yet settled (e.g., recruiting, not yet recruiting, ..). This leaves us with 164,326 protocols, which we split into train, validation and test sets with ratios of 70%, 15% and 15%, respectively. In order to assign risk-related labels to the CT's, we consider those with "completed" status as low-risk and otherwise consider them as high-risk. Before proceeding with graph formation as described above, we eliminate all label-sensitive content from the protocols (status field, results field, ..).

Classification results: The results of binary classification on the test set of the CTGov collection described above are presented in Table 2 for the BOW-based featurization approach, and in Table 3 for the transformer-based featurization, where we used the BERT-like language model described in [15], which is pre-trained on a large collection of biomedical text.

		precision		recall		f1-score		AUC	
		micro	macro	micro	macro	micro	macro	ROC	PR
GCN	empty	0.8455	0.8064	0.8455	0.8157	0.8455	0.8108	0.8990	0.8958
	cnd.+int.	0.8502	0.8128	0.8502	0.8179	0.8502	0.8153	0.9006	0.8974
	cnd.+int.+ph.	0.8634	0.8337	0.8634	0.8210	0.8634	0.8270	0.9050	0.9017
GAT	empty	0.8597	0.8283	0.8597	0.8175	0.8597	0.8226	0.9008	0.8989
	cnd.+int.	0.8635	0.8283	0.8635	0.8377	0.8635	0.8327	0.9158	0.9127
	cnd.+int.+ph.	0.8615	0.8297	0.8615	0.8222	0.8615	0.8258	0.9033	0.9011
SAGE	empty	0.8688	0.8395	0.8688	0.8301	0.8688	0.8346	0.9061	0.9048
	cnd.+int.	0.8753	0.8538	0.8753	0.8278	0.8753	0.8392	0.9080	0.9064
	cnd.+int.+ph.	0.8759	0.8537	0.8759	0.8300	0.8759	0.8405	0.9087	0.9076

Table 2: Binary classification under different connecting graph configurations and graph convolutional layers. Results based on Bag-Of-Words features.

As it can be seen from the results of Table 2 and 3, for both cases and under all graph convolutional layers, the new edges induced by the introduction of the connecting graphs improves the performance over the baseline "empty" graph. Note that while this improvement is not drastic in this case, because essentially no new source of information has been added, the proposed framework can be highly beneficial when external sources are available that are not straightforward to represent as vectorial features.

As a general conclusion, the use of the super-graph topological features seems to consistently increase the predictive power of the models. This is consistent with works in other domains that show the benefit of topological features in the predictive power of machine learning models based on graph abstractions [14].

[7] Source code at https://github.com/sssohrab/ct-classification-graphs.

[8] https://ClinicalTrials.gov/AllAPIJSON.zip

		precision		recall		f1-score		AUC	
		micro	macro	micro	macro	micro	macro	ROC	PR
GCN	empty	0.8739	0.8476	0.8739	0.8333	0.8739	0.8399	0.9105	0.9095
	cnd.+int.	0.8721	0.8430	0.8721	0.8359	0.8721	0.8393	0.9119	0.9105
	cnd.+int.+ph.	0.8778	0.8542	0.8778	0.8356	0.8778	0.8441	0.9146	0.9129
GAT	empty	0.8647	0.8306	0.8647	0.8357	0.8647	0.8331	0.9118	0.9110
	cnd.+int.	0.8668	0.8324	0.8668	0.8409	0.8668	0.8364	0.9164	0.9148
	cnd.+int.+ph.	0.8641	0.8288	0.8641	0.8395	0.8641	0.8338	0.9155	0.9139
SAGE	empty	0.8666	0.8326	0.8666	0.8388	0.8666	0.8356	0.9147	0.9139
	cnd.+int.	0.8804	0.8520	0.8804	0.8495	0.8804	0.8507	0.9258	0.9238
	cnd.+int.+ph.	0.8736	0.8424	0.8736	0.8439	0.8736	0.8431	0.9202	0.9189

Table 3: Binary classification under different connecting graph configurations and graph convolutional layers. Results based on pre-trained transformer text featurizers.

5 Discussions

The literature of CT studies identifies various common reasons behind the very frequent scenario of trial failures (see e.g., [8] [27]). While in general, it is useful to know the common reasons behind trial failure on the average, it would perhaps be much more beneficial to be able to predict the outcome of any given trial study, and before its execution. This can be an important step towards optimization of trial design to mitigate the risk factors and eventually to increase the odds of success. Given the central importance of CT's within the whole drug design pipeline, any such risk mitigation can have direct impact on medication prices and their time-to-market.

In this work, relying on large-scale data and machine learning techniques, we proposed one such framework to predict CT behavior from raw protocols. While we showed very high risk classification performance, it should be mentioned, however, that our measure of trial success in this work, i.e., the reported completion vs. non-completion of trials is perhaps rather simplistic. For a more realistic risk quantification and subsequently risk prediction for candidate trials, more detailed criteria like the duration of the study, the attrition rate of patients, or the toxicity reports of drugs should be taken into account. This, however, requires more data resources that are usually not publicly available in large scale.

On another note, our adopted methodology is highly flexible and can benefit from various sources of information while treating them under one common framework, i.e., deep learning using GNN's. This is thanks to the very versatile structure of graphs that can incorporate both vectorial features and topological information. In particular, our main proposition in this work, i.e., to enrich graph classification by using further connectivity information can be very suitable for using external data resources. While we used attributes like trial phase and medical intervention that are already present within the CT protocols, future work can consider external medical onthologies or drug information as better similarity measures to connect CT's.

6 Conclusions

With the increasing popularity of graph-based machine learning approaches within the healthcare domain, this work investigated the role of the problem formulation and the way the graph is constructed on the overall task performance. For our application example of CT protocol classification, we showed that while they can be considered as independent graph objects and hence formulated under a graph classification problem, by connecting the objects to one single large graph using some domain-aware similarity measures and hence formulating the problem as node classification, consistent performance gains can be achieved. This can particularly be useful for cases where some extra metadata is available that cannot be directly encoded as features. Our experiments were performed on the publicly CTGov data, for which we provide the open source codes.

References

1. Bronstein, M.M., Bruna, J., Cohen, T., Veličković, P.: Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. arXiv preprint arXiv:2104.13478 (2021)
2. Chiang, W.L., Liu, X., Si, S., Li, Y., Bengio, S., Hsieh, C.J.: Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 257–266 (2019)
3. Choi, E., Bahadori, M.T., Song, L., Stewart, W.F., Sun, J.: Gram: graph-based attention model for healthcare representation learning. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 787–795 (2017)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Elkin, M.E., Zhu, X.: Predictive modeling of clinical trial terminations using feature engineering and embedding learning. Scientific reports **11**(1), 1–12 (2021)
6. Elkin, M.E., Zhu, X.: Understanding and predicting covid-19 clinical trial completion vs. cessation. Plos one **16**(7), e0253789 (2021)
7. Ferdowsi, S., Borissov, N., Knafou, J., Amini, P., Teodoro, D.: Classification of hierarchical text using geometric deep learning: the case of clinical trials corpus. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing
8. Fogel, D.B.: Factors associated with clinical trials that fail and opportunities for improving the likelihood of success: a review. Contemporary clinical trials communications **11**, 156–164 (2018)
9. Follett, L., Geletta, S., Laugerman, M.: Quantifying risk associated with clinical trial termination: a text mining approach. Information Processing & Management pp. 516–525 (2019)
10. Gainza, P., Sverrisson, F., Monti, F., Rodola, E., Boscaini, D., Bronstein, M., Correia, B.: Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. Nature Methods **17**(2), 184–192 (2020)
11. Geletta, S., Follett, L., Laugerman, M.: Latent dirichlet allocation in predicting clinical trial failures (2019)
12. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: International Conference on Machine Learning. pp. 1263–1272. PMLR (2017)
13. Glynn, E.F., Hoffman, M.A.: Heterogeneity introduced by ehr system implementation in a de-identified data resource from 100 non-affiliated organizations. JAMIA open **2**(4), 554–561 (2019)

14. Gouareb, R., Can, F., Ferdowsi, S., Teodoro, D.: Vessel destination prediction using a graph-based machine learning model. In: Proceedings of International School and Conference on Network Science. Porto, Portugal (2022)
15. Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., Poon, H.: Domain-specific language model pretraining for biomedical natural language processing (2020)
16. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 1025–1035 (2017)
17. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
18. Martin, L., Hutchens, M., Hawkins, C.: Trial watch: clinical trial cycle times continue to increase despite industry efforts. Nature Reviews Drug Discovery **16**(3), 157–158 (2017)
19. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3973–3983 (2019)
20. Shang, J., Ma, T., Xiao, C., Sun, J.: Pre-training of graph augmented transformers for medication recommendation. arXiv preprint arXiv:1906.00346 (2019)
21. Stokes, J.M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N.M., MacNair, C.R., French, S., Carfrae, L.A., Bloom-Ackermann, Z., et al.: A deep learning approach to antibiotic discovery. Cell **180**(4), 688–702 (2020)
22. Teodoro, D., Pasche, E., Gobeill, J., Emonet, S., Ruch, P., Lovis, C.: Building a transnational biosurveillance network using semantic web technologies: requirements, design, and preliminary evaluation. Journal of medical Internet research **14**(3), e73 (2012)
23. Teodoro, D., Sundvall, E., João Junior, M., Ruch, P., Miranda Freire, S.: Orbda: An open ehr benchmark dataset for performance assessment of electronic health record servers. PloS one **13**(1), e0190028 (2018)
24. Teodoro, D.H., Choquet, R., Schober, D., Mels, G., Pasche, E., Ruch, P., Lovis, C.: Interoperability driven integration of biomedical data sources. Studies in health technology and informatics **169**, 185–9 (2011)
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
26. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
27. Williams, R.J., Tse, T., DiPiazza, K., Zarin, D.A.: Terminated trials in the clinicaltrials.gov results database: Evaluation of availability of primary outcome data and reasons for termination. PLOS ONE **10**(5), 1–12 (05 2015). https://doi.org/10.1371/journal.pone.0127242, `https://doi.org/10.1371/journal.pone.0127242`
28. Wong, C.H., Siah, K.W., Lo, A.W.: Estimation of clinical trial success rates and related parameters. Biostatistics **20**(2), 273–286 (2019)
29. Wouters, O., McKee, M., Luyten, J.: Estimated research and development investment needed to bring a new medicine to market, 2009-2018 [published march 3, 2020]. JAMA
30. Zeng, H., Zhang, M., Xia, Y., Srivastava, A., Malevich, A., Kannan, R., Prasanna, V., Jin, L., Chen, R.: Deep graph neural networks with shallow subgraph samplers. arXiv preprint arXiv:2012.01380 (2020)
31. Zhu, W., Razavian, N.: Variationally regularized graph-based representation learning for electronic health records. In: Proceedings of the Conference on Health, Inference, and Learning. p. 1–13. CHIL '21, Association for Computing Machinery, New York, NY, USA (2021)