A Citations in medical research

Medical research communication follows a standardised format and occurs primarily through publication [**BMCMedicalResea** A component of that research is a literature review that contains established facts about a topic. These established facts are corroborated with references to their source of evidence. Citations adhere to a standard format and are typically listed at the end of the research article.

It follows, then, that relations between research can be elicited from the analysis of these ciations, with the assumption being that studies that cite, or are cited, by a research article are relevant to that research.

A.1 Improvements in encoder performance using dzirect citations

Medical CAL TAR has already benefited from citation utilisation. To date, the most performant encoder model ($BioLinkBERT_{base}$) [**yasunaga2022linkbertpretraininglanguagemodels**] on the CLEF dataset in a CAL setting, leveraged citations between research to achieve state-of-the-art within the CAL process [**goharian˙reproducibility˙2024**]. The *LinkBERT* approach was to view a pertaining corpus as a graph of documents, with each document being a vertex and hyperlinks forming edges between documents. These linked documents were then placed within the same context, which was different from that of *BERT* random document allocation, in which no linkage between documents within a context window is required. *LinkBERT* differs from curriculum learning, where a model is provided with examples of increasing difficulty, as the context windows were not ordered by difficulty. A domain-specific variant of *LinkBERT*, *BioLinkBERT*, was created, which was pretrained only on PubMed articles, with linkage of documents being determined through citations of that research[1]. Models were then trained using standard masked language modelling and next-sentence prediction. The performance of a base model (100M parameters) and a large model (340M parameters) were compared to *PubMedBert*[2] in BLURB[**guDomainSpecificLanguageModel2021**], MedQA-USMLE[**jinWhatDiseaseDoes2020**], and MMLU-professional medicine (medical-specific downstream benchmark tasks)[**hendrycksMeasuringMassiveMultitask2021**]. $BioLinkBERT_{large}$ achieved state-of-the-art on all reported benchmarks, with an improvement in the BLURB score of 3.2% above PubMedBERT.

In the previously reported Goldilock Reproduce study, where an Enocder CAL approach was used, $BioLinkBERT_{base}$ formed the classifier. This author recreated their experiment with $BioBERT_{large}$[3] as a classifier model and achieved higher performance in R-Precision in 7 of 12 datasets/policy combinations. The Friedman test for individual datasets found significant differences between the FPT epochs 4 out of 12 times. However, when considering all datasets together, there was no significant difference between the FPT epochs and R-precision for relancy selection policy or uncertainty selection policy. This indicates that the "Goldilocks problem", which was previously reported in the literature in non-medical domains was not apparent when using the large BiolinkBERT model for the CLEF dataset within a CAL process, indicating that further pretraining of models was unneccesary as it does not produce an statistically significant improvement in R-precision. The average R-precision of each FPT epoch is reported in Table 2, with the highest R-precision for relevancy selection policy being 0.847 at further pretrain epoch 2 and the highest R-precision for uncertainty being 0.832 at further pretrain epoch 1.

Key findings from this research is that an optimal pretraining epoch is unlikely to be found within the CLEF dataset and hence not a viable avenue for future research. In terms of experimental design, certain hyperparameters were chosen without clear reasoning (such as batch size being 25, fine tuning for 20 epochs and stopping after 501 documents labelled). This limitation is thought to be a barrier to improving the performance of the encoder CAL process within that experimental framework, given that reported R-Precision values are already close to the natural

[1] https://huggingface.co/michiyasunaga/BioLinkBERT-base

[2] https://huggingface.co/microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext

[3] https://huggingface.co/michiyasunaga/BioLinkBERT-large

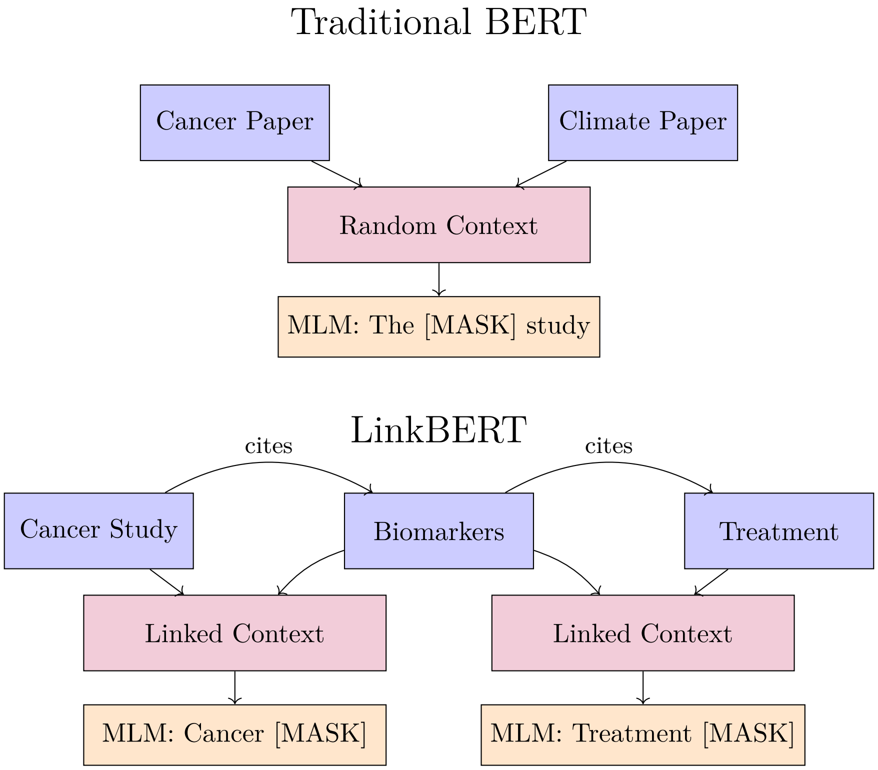


Figure 1: Comparison of document processing in traditional BERT versus LinkBERT. Traditional BERT (top) randomly groups documents into context windows, while LinkBERT (bottom) uses citation relationships to create meaningful document groupings for pretraining. The citation-based grouping ensures that semantically related documents are processed together during masked language modeling tasks.

ceiling of 1 (with R-Precision reaching 0.945 in some cases). In datasets where there is worse performance, (0.82 R-precision for relevancy, 0.791 R-precision for uncertainty in the CLEF 2019 dataset) comprehension of why is obsfucated by the use of LLMs. Furturemore, using a more performance/larger model is likely to be fruitful future research, however it is dependent on the availability and development of superior models. This undertaken research however highlight that leveraging citations themselves was valuable to the CAL process.

A fundamental question emerges from this research: Is contextual understanding of references truly necessary for effective CAL? Several factors suggest that citation networks alone might be sufficient and potentially superior. First, citations themselves represent a form of knowledge distillation, where domain experts have already identified meaningful relationships between documents. Second, analysing reference networks is computationally more efficient than processing full textual contexts. Third, citation network models tend to be more stable when updated, compared to contextual models. Fourth, the contextualization of citation networks may actually introduce noise into what would otherwise be clear citation signals.

While *BioLinkBERT* represents a sophisticated approach that combines citation networks with contextual language understanding, this integration presents both advantages and limitations. The model's ability to capture complex semantic relationships between documents is valuable, but the contextual processing introduces potential inefficiencies. During pretraining, when linked documents are placed in the same context, the model must process all content within those documents—including sections that may be tangential or unrelated to the citing paper's specific reference. This contextual noise could potentially dilute the precision of the more direct relationships that citations inherently represent. In contrast, pure citation links directly capture intentional scholarly connections made by domain experts, providing a cleaner signal without the additional complexity of processing potentially irrelevant contextual information.

Collection	Dataset size	Model	R-Precision (↑)		Friedman (p)	
			Rel.	Unc.	Rel.	Unc.
Clef 2019 dta test	8	BiolinkBert-Base-ep0	**0.909**	**0.857**	—	
		BiolinkBert-Large-ep0	0.897	0.803		
		BiolinkBert-Large-ep1	0.827	0.832		
		BiolinkBert-Large-ep2	0.812	0.774	0.914	0.632
		BiolinkBert-Large-ep5	0.841	0.814		
		BiolinkBert-Large-ep10	0.881	0.846		
Clef 2017 test	30	BiolinkBert-Base-ep0	0.812	0.794	—	
		BiolinkBert-Large-ep0	0.828	0.797		
		BiolinkBert-Large-ep1	0.826	**0.827**		
		BiolinkBert-Large-ep2	**0.858**	0.804	**<0.05**	**<0.05**
		BiolinkBert-Large-ep5	0.827	0.777		
		BiolinkBert-Large-ep10	0.799	0.757		
Clef 2017 train	20	BiolinkBert-Base-ep0	**0.838**	0.761	—	
		BiolinkBert-Large-ep0	0.778	0.765		
		BiolinkBert-Large-ep1	0.808	0.789		
		BiolinkBert-Large-ep2	0.767	0.701	**<0.05**	0.28
		BiolinkBert-Large-ep5	0.816	0.786		
		BiolinkBert-Large-ep10	0.827	**0.796**		
Clef 2018 test	30	BiolinkBert-Base-ep0	0.794	0.780	—	
		BiolinkBert-Large-ep0	0.789	0.774		
		BiolinkBert-Large-ep1	**0.812**	0.790		
		BiolinkBert-Large-ep2	0.797	**0.791**	0.52	0.50
		BiolinkBert-Large-ep5	0.763	0.773		
		BiolinkBert-Large-ep10	0.763	0.769		
Clef 2019 DTA int. train	20	BiolinkBert-Base-ep0	0.939	0.923	—	
		BiolinkBert-Large-ep0	0.939	0.902		
		BiolinkBert-Large-ep1	0.941	0.935		
		BiolinkBert-Large-ep2	0.948	0.921	0.78	0.50
		BiolinkBert-Large-ep5	0.952	0.945		
		BiolinkBert-Large-ep10	**0.945**	**0.947**		
Clef 2019 DTA int. test	20	BiolinkBert-Base-ep0	**0.934**	**0.900**	—	
		BiolinkBert-Large-ep0	0.899	0.856		
		BiolinkBert-Large-ep1	0.904	0.840		
		BiolinkBert-Large-ep2	0.909	0.878	0.87	**<0.05**
		BiolinkBert-Large-ep5	0.882	0.835		
		BiolinkBert-Large-ep10	0.865	0.841		

Table 1: Performance comparison across different collections and models

Table 2: Average R-precision of each FPT epoch for CLEF dataset

Policy	ep0	ep1	ep2	ep5	ep10
Uncertainty	0.813	0.832	0.813	0.815	0.814
Relevancy	0.840	0.845	0.847	0.842	0.835

A.2 Direct citation network mining within medicine research

Despite these more complex citation network mining approaches being somewhat performant, simpler, more robust approaches to citation network mining already exist within medicial research. Let G be a citation graph where:

- D_i represents a research article of interest as a vertex in G
- D_{ip} represents the set of articles referenced by D_i

- D_{if} represents the set of articles that reference D_i
- Both sets are subsets of G: $D_{ip}, D_{if} \subset G$
- $D_{ip} \cap D_{if} = \emptyset$, so searching both sets will provide different relevant articles

Relevancy is defined as a function $R : D \to [0, 1]$, where:

- 0 denotes no relevance
- 1 denotes maximum relevance
- For any set of documents D_{set}, relevancy is defined as $R(D_{set}) = R(d)|d \in D_{set}$

Two primary search approaches are defined:

- Backward citation searching (BCS): examining all articles in D_{ip}[**lefebvre2011cochrane**, **akers2009systematic**]
- Forward citation searching (FCS): examining all articles in D_{if}*[4]

These approaches are both simple and effective, inherently respecting temporal relationships between research articles (i.e., you cannot cite a paper that has yet to be published). More so, BCS and FCS are recommended to be used in the identification phase of Cochrane systematic review generation, with 87% of Cochrane reviews between November 2016 to January 2017 reporting the use of BCS and 9% reporting the use of FCS[**briscoeConductReportingCitation2019**]. Recommendations regarding BCS/FCS in the the search stage are found within the Cochrane Handbook, with BCS being mandatory (C30), and FCS not being mentioned [**MECIRManualCochrane**]. No such recommendations exist for BCS and FCS use within the screening phase within the Cochrane Handbook. Additionally, there is a paucity of research outlining the use of BCS and FCS within the active learning process when searching multiple citation indexes (Search Strategy outlined in Figure 2). An important clarification is required to understand the novality of this approach. As previously mentioned, this PhD is focusing on the title and abstract phase within the screening phase of systematic review generation. BCS and FCS is traditionally used in the identification phase only (see Figure **??** for overview of phases of the systematic review process), with title and abstracts being screened to reduce the work required in the more costly full text screening phase. This is not a limiting factor for comptuation, and hence restriction of this phase to just title and abstracts is redundant. This amendment to the screening process is also not intended to remove the full text screening phase, as each phase has a different requirement, with title and abstract screening being focused only on relevancy to the topic, and full text screening being focused on quality and eligibility of the research.

[4] FCS involves using a citation index to identify studies that cite a source study. A citation index is a database of scholarly articles and their citations, such as PubMed, Google Scholar, Scopus or OpenAlex

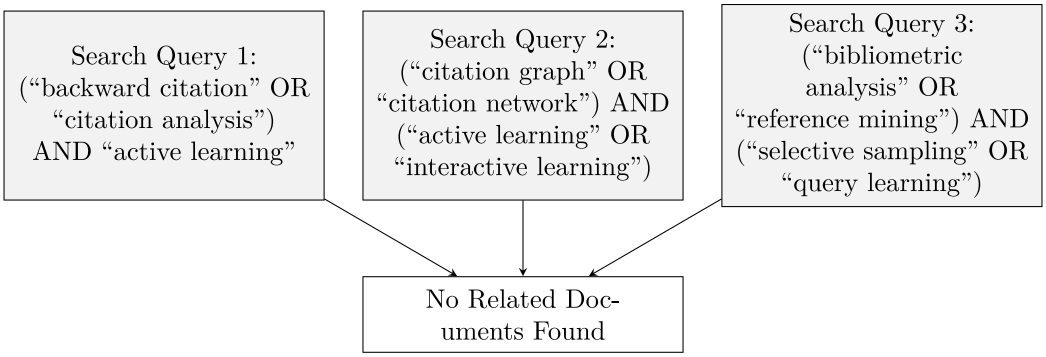
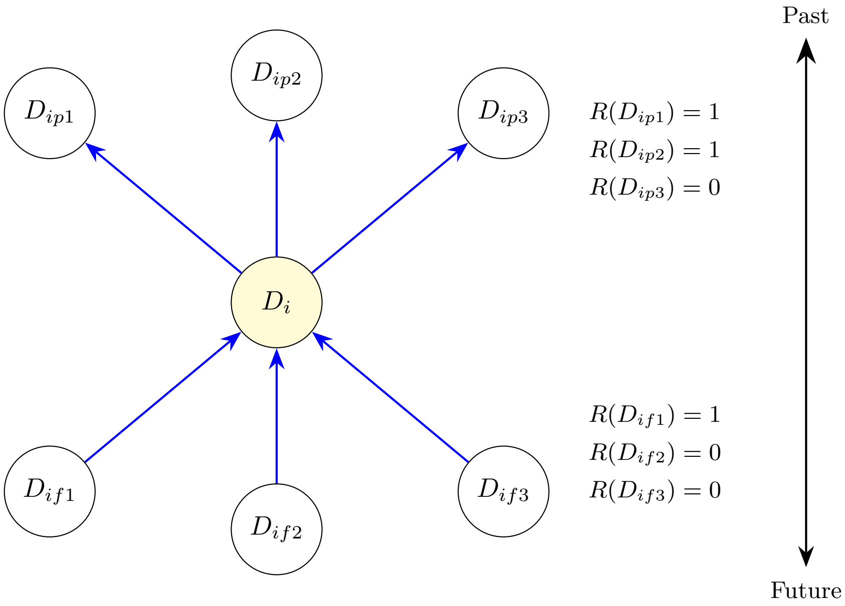


Figure 2: Results from literature search on citation index arxiv and pubmed demonstrating absence of related works, ran on 13th November 2024



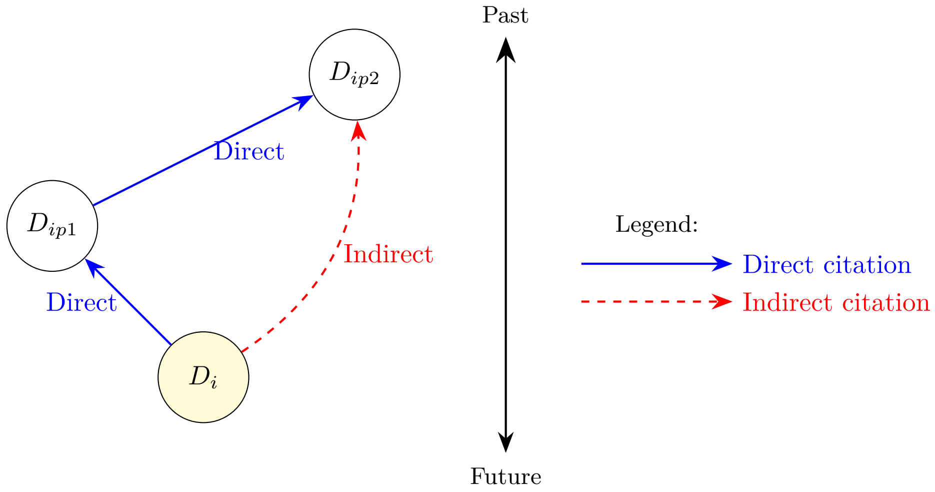
While the citation network mining approach could be used to group articles of interest within a context window, similar to that of *BioLinkBERT* (which used FCS, not BCS), pretraining a model from scratch using this approach is obviously outside the possibility of this PhD project, due to hardware constraints. However, citation network mining can be used to augment the CAL process in ways that overcome some of the limitatitions of this process.

Firstly, CAL requires labelled data to train a classifier model, which is assumed to perform better with more data points. Encoder CAL approaches suffer disproportionately to that of feature-based CAL approaches due to their need for larger amounts of training data to effectively learn meaningful representations. This is because encoder models like BERT need to learn complex contextual relationships between words and concepts, whereas feature-based models can rely on simpler statistical patterns. When working with limited labeled data in the early stages of screening, encoder models may struggle to generalise well, potentially leading to suboptimal performance in identifying relevant documents. As discussed in the Encoder CAL process, often a single sample seed document is used during the first epoch for fine-tuning. A better approach would be to exhaust the citation network of that seed document first for labelling, before using revealed relevant documents to fine-tune the model, potentially resulting in a more performant model at the earlier stages of screening with less oracle cost. This potentially overcomes limitations of the Encoder CAL process, and also reduces computational costs (as citation network mining is less computationally expensive than fine-tuning).

Early work performed by the Author on a subset of the CLEF dataset demonstrates that exhausting the BCS network of a seed document resulted in greater R-Precision values when compared to that of the Encoder CAL process at similar points. The logical, and simple augmentation of the encoder CAL approach would be to exhaust both BCS and FCS networks of a seed document prior to initiating the encoder CAL process.

A.3 Extending current citation network mining approaches

One of the issues with citation network mining is that it is not able to identify indirect citations. For example, if D_i references D_{ip1} and D_{ip1} references D_{ip2}, then D_i and D_{ip2} are related, even though D_i does not cite D_{ip2} directly. This limitation would prevent the use of citation network mining to replace the encoder CAL process entirely. Other citation network mining approaches have been proposed to address this issue:



Section about adding other features to the network, i.e. Author, topic

A.4 Graph Neural Networks

A.5 LLMs and citation network mining

The motivation for using LLMs and graph networks is to combine the structurality of graph citation networks with the ability of the LLMs to comprehend the semantic meaning of documents. As outlined, citation network graph analysis occurs above the document level through utilisation of extracted features about documents. LLMs are a natural replacement for extraction of features, as they possess the ability to understand semantic meaning of documents. The ultimate goal to use LLMs and graph networks is to complment and enhance isuses with the other.

Research has been conducted into the use of LLMs within the graph neural networks, and has developed a robust taxonomy for categorising the use of LLMs within graph networks [**llm4g**].

The first application of LLms within graph networks is to use LLM as an enchancer. Typically graph neural networks encode text into nodes using simple bag-of-words, skip-gram or TF-IDF. LLMs are able to encode text into nodes using more complex features, such as semantic meaning, which can be used within the graph neural networks. This can be further subdivided into explanation based and embedding based enchancers.

Explanation based enhancers query an LLM using prompting to capture higher level features about documents, which is used to enrich node representations prior to processing with a graph neural network, with the process being abstracted in in Figure 3. The approach used by https://arxiv.org/pdf/2305.19523 was to prompt GPT 3-5 with the abstract and text of a document along with a questions about that document using a zero-shot approach. The LLM reponse then forms features which are amended to the original node representations. Issues with this approach is that this requires domain specific knowledge, as features which are deemed important (and hence prompt used) are dependent on the domain of the research. It was performant on the pubmed domain, scoring greater node classification accuracy using this approach (0.9618 ± 0.0053) than utilising an LLM alone on pubmed data (0.9494 ± 0.0046).

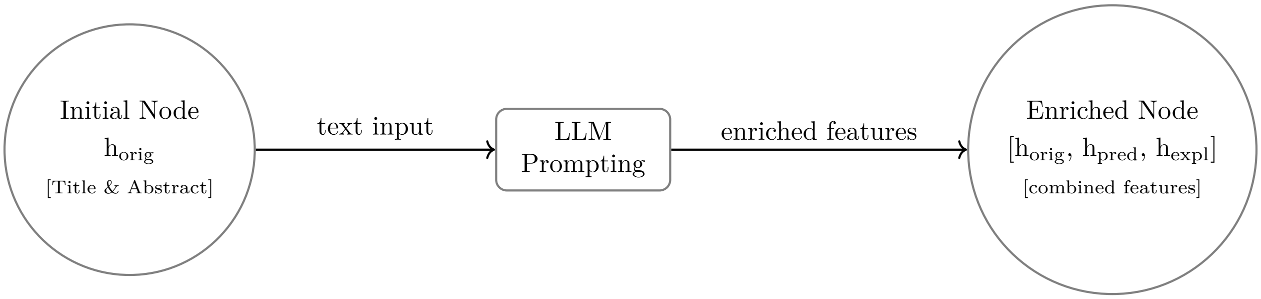


Figure 3: Node feature enrichment process using LLM and LM

B Notes on Graph Neural Networks

A node is represented by a feature matrix, which contains information about the document. This **Node feature matrix**, X, which has the dimensions of m (the number of nodes) and n (the number of features). $X \in \mathbb{R}^{m \times n}$. X does not have to be a square matrix, and does not encode any information about the structure of the graph.

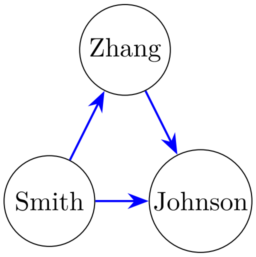
Consider 3 research papers as nodes, with features: [Author, Title Length, Abstract Length, Citation Count]

$$X = \begin{bmatrix} \text{”Smith”} & 82 & 500 & 45 \\ \text{”Johnson”} & 95 & 475 & 23 \\ \text{”Zhang”} & 67 & 612 & 89 \end{bmatrix}$$ Where $X \in \mathbb{R}^{3 \times 4}$ represents:

3 papers (rows) 4 features per paper (columns) Mixed data types (categorical and numerical)

Structural information is encoded in the **adjacency matrix**, A, which has the dimensions of m (the number of nodes) and m (the number of nodes). $A \in \mathbb{R}^{m \times m}$. A encodes information about the structure of the graph, and is used to determine relationships between nodes. Conventionally the source nodes are the rows, and the destination nodes the columns of the matrix. 1 indicates an edge between the source node u and destination node v. Note that there is a choice to make here, with the diagonal of the matrix being 0 or 1. This choice is based on wheter you consider the source node to be connected to itself. In cases where the representation of the node is dependent on itself and adjacent nodes, the diagonal should be set to 1. In the scenario of citation networks, the diagonal should be set to 1, as a paper is likely to reference and build upon its own findings throughout. By setting the diagonal to 0, it is akin to attempting to predict the representation of the node base only on its adjacent nodes, which is not the case in citation networks. If an adjaceny metrix is symmetric around it's diagonal, then the graph is undirected, otherwise it is directed (i.e. U is connected to V and V is conencted to U). In citation networks, this is not the case, as because paper A cites paper B, it does not mean the reverse is true.

Consider the same 3 research papers, with the following adjacency matrix: $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ Which represents the following graph:



With both X and A defined, we can numerically represent the graph. The node feature matrix X is the initial/input node features, with our goal for learning on graphs to learn node embeddings $H \in \mathbb{R}^{N \times D}$ where D is a chosen hidden dimension size.

C Message Passing Neural Networks

We need an approach that can work with the graph structure, which has variable number of nodes and edge conenctions between nodes. Historically with the CNN architecture, the input size was fixed, and the network was able to learn spatial invariance through the use of convolutional filters that were invariant to the location of the feature in the input. With graph structured data, the number of nodes and connections between nodes can vary for each graph, and spatial invariance is not invariant to the location of the feature in the graph.

Message Passing Neural Networks (MPNNs) are a type of graph neural network that can learn spatial invariance through the use of message passing between nodes. The basic idea of MPNNs is to iteratively update node representations by passing messages between connected nodes. This process is repeated for a fixed number of iterations, or until convergence.

The process is defined as follows:

- Message: every node decides how to send information to neighboring nodes it is conenected to by edges
- Aggregate: nodes recieve messages from all their neighbors, who also passed messages and decides how to combine the information from all of its neighbors.
- Update: each node decies how to combine neighbourhood information with its own information and updates it embedding for the next timestep.

By doing this we have nodes pass each other information and disseminate information around the graph, allowing the network to learn spatial invariance. This can be repeated for a fixed number of iterations (K), with the larger the value of K, the more the more diffuse the information around the graph becomes.

Each section of the MPNN process in more detail:

C.1 Message

The source node U will pass a message m_{uv} to the destination node V. The message depends on the GNN architecture with the easiest example message being passed being U node's feature h_u vector to V.

C.2 Aggregate

The destination node V will recieve messages from all its neighbouring nodes, and needs to decide how to combine the information from all of its neighbours. This is typically done using a sum, average or max pooling of the messages from all neighbouring nodes. It is important that the aggregation function has to be a permutation invariant function, as the order of the messages should not affect the output.

This gives us a combined neighbourhood node embedding, denoted as $h_{N(V)}$, where $N(V)$ is the set of all neighbouring nodes to V, meaning all nodes connected to V by an edge.

$h_{N(v)}^{k+1} = AGGREGATE(h_u^k, \forall u \in N(v))$

C.3 Update

Each node updates its own embedding based on the combined neighbourhood embedding and its own embedding from the previous timestep.

$h_v^{k+1} = \sigma(W \cdot CONCAT(h_v^k, h_{N(v)}^{k+1}))$