

ALMA MATER STUDIORUM – UNIVERSITÀ DI
BOLOGNA

CAMPUS OF BOLOGNA
Computer Science and Engineering - DISI
Master Degree in Artificial Intelligence

INFUSING STRUCTURED MEDICAL KNOWLEDGE INTO
LANGUAGE MODELS: GRAPH NEURAL PROMPTING FOR
HEALTHCARE QUESTION ANSWERING

Supervisor:

Prof. Gianluca Moro

Co-Supervisors:

Dott. Giacomo Frisoni

Dott. Lorenzo Valgimigli

Candidate:

Yiran Zeng

Third Session
Academic Year 2022 – 2023

KEYWORDS

Natural Language Processing

Graph Neural Prompting

Structured Knowledge

Question Answering

Language Models

但行好事，莫问前程。
- 《增广贤文》

*If you shed tears when you miss the sun,
you also miss the stars.
- Rabindranath Tagore*

Abstract

Current natural language processing models have made significant strides in understanding medical texts. Recent research leverages non-parametric external sources of relevant factual information. It taps into latent world knowledge to enhance performance and interpretability in the medical domain. However, a lack of in-depth understanding of structured medical knowledge limits its practical application in complex medical contexts.

In this thesis, we present MEDGNP, the first method for infusing structured medical knowledge into language models, a plug-and-play new approach applying a prompting framework by grounding knowledge to enhance models' capabilities in addressing question-answering tasks within the medical domain. MEDGNP incorporates various designs, including a graph attention network, cross-modal attention module, domain projector, and a bilinear model for self-supervised link prediction.

We evaluated our method on three popular medical question-answering datasets: PubMedQA, BioASQ and MedQA. After knowledge grounding using MEDGNP method, we get improvement across all datasets from 1.26% to 6.0% by leveraging 30M extra parameters, which takes only 12% of the original model. Furthermore, MEDGNP allows a generalist model to achieve comparable results with state-of-the-art models that underwent a domain-specific pre-training extension while using up to 12x fewer trainable parameters, proving the effectiveness of this approach to successfully knowledge grounding language model. Finally, we discuss potential future work on this topic.

Introduction

Motivation and contribution

Natural Language Processing (NLP) plays a pivotal role in medicine, with extensive applications and immense potential. NLP not only facilitates the deep analysis of medical texts through its Natural Language Understanding (NLU) capabilities but also boasts Natural Language Generation (NLG) abilities, making it suitable for answering medical questions and providing relevant information and recommendations [1]. In recent years, with the rapid advancement of deep learning, many external knowledge sources have been introduced to enhance pre-trained Language Models (LMs). This enhancement enables these models to handle knowledge-intensive NLP tasks more effectively [2]. This progress has ushered in new opportunities and possibilities for addressing complex tasks in the medical domain.

However, despite the remarkable strides made by NLP in the medical field, most current research and applications still predominantly rely on biomedical literature, such as PubMed and PMC, as databases [3], often neglecting the valuable knowledge that can be found within structured data. It should be noted that the medical field possesses extensive structured databases, such as the Unified Medical Language System (UMLS). These databases contain rich information on standardised terms and concepts in various medical domains [4]. Currently, the utilisation of these structured databases has yielded significant results in multiple tasks, ranging from Named Entity Recognition (NER) tasks [5, 6, 7] to Synonymy Prediction tasks [8, 9, 10]. In this era of medical information, integrating structured medical knowledge into the training process of NLP models is an exciting and potentially revolutionary direction.

To address this challenge and unlock the potential of structured medical knowledge, we propose an innovative approach - MEDGNP. Our approach aims to embed structured medical knowledge into pre-trained LMs, thereby assisting pre-trained models in learning useful information from external knowledge. Specifically, MEDGNP begins by encoding complex graph knowledge into concept embedding through a graph encoder. Then, it uses a cross-modal

attention module to identify the most relevant nodes about the question [11]. Finally, a domain projector bridges the inherent disparities between the graph and text domains. After that, we integrate all node embedding into a single graph-level embedding. By incorporating graph prompting techniques, we can convey the resulting graph-level embedding to the LMs, enabling them to utilise external grounded knowledge more effectively to accomplish Question Answering (QA) task in medical domain.

This task blends open-domain QA and knowledge-intensive challenges, demanding comprehensive medical knowledge and complex terminology understanding. The absence of a gold-standard context for each question in this task requires extracting and interpreting relevant information from extensive, unstructured sources, significantly increasing difficulty and necessitating advanced NLP techniques for precise answers.

Our goal is to present significant performance improvements in the QA tasks within the medical domain. We believe that our approach not only mitigates the issue of increased dependence on training resources due to the growth in model size but also enables the model to access external knowledge in real-time, thereby enhancing performance across various applications in the medical field. In the following sections, we will provide a detailed account of our method, experimental design, and results, to demonstrate the efficacy and potential value of this approach.

Thesis Organisation

The thesis is organised as follows:

- **Chapter 1** - Theoretical Framework: presents a general framework on the concepts of Knowledge Graph Partitioning and Retrieval-enhanced Models, etc. Furthermore, it introduces UMLS.
- **Chapter 2** - Related Works: clarify the context and summarise prior work related to this project.
- **Chapter 3** - Methods: delve into the inner workings of our proposed solution, explaining its key components and operational processes in a step-by-step manner through a top-down approach.
- **Chapter 4** - Experimental Setup and Results: our experimental setup and discussion of the results.

Contents

1	Theoretical Framework	1
1.1	Language Models	1
1.2	Knowledge Graph Partitioning	3
1.3	Retrieval-Enhanced Model	4
1.4	Prompt Tuning	5
1.5	UMLS	7
2	Related Work	9
2.1	Knowledge Graph Embedding	9
2.1.1	Translation-Based Models	9
2.1.2	Embedding-Based Models	10
2.1.3	Neural Network-Based Models	11
2.1.4	Integration with Our Work	12
2.2	Knowledge-Enhanced Language Models	13
2.2.1	Retrieval-Augmented Approaches	15
2.2.2	Deep Integration Techniques	16
2.2.3	Integration with Our Work	18
3	Method	19
3.1	Overview	19
3.2	Subgraph Retrieval	20
3.3	MEDGNP	21
3.3.1	Graph Encoder	22
3.3.2	Self-Attention Module	22
3.3.3	Text Embedding and Transformation	23
3.3.4	Cross-Modality Attention Module	23
3.3.5	Pooling and Domain Projector	24
3.3.6	Self-Supervised Link Prediction	24
3.4	Training with Language Model	25
4	Experiments and Results	26
4.1	Experimental Setup	26

4.1.1	Data Preparation	26
4.1.2	Model configuration	27
4.1.3	Implementation Details	27
4.2	Results	29
4.2.1	Performance Comparison	29
4.2.2	GAT Design Comparison	31
4.3	Ablation Study	32
	Conclusions and Future Challenges	34
	Acknowledgments	36
	Bibliography	38
	Appendix	49

List of Figures

1.1	The transformer model architecture. It consist of two separate modules: encoder and decoder.	2
1.2	Illustrate an example of KG partitioning. Different colours represent different categories of nodes; this partitioning operates based on the types of nodes.	3
1.3	Text generation example based on retrieval enhancement.	5
1.4	Example of prompt-tuning with zero-shot learning.	6
1.5	Example of prompt-tuning with one-shot learning	6
1.6	Comparison between fine-tuning and prompt-tuning. In the figure, the circles represent the pre-trained LM, and the rectangular boxes represent various downstream NLP tasks.	7
2.1	Illustrate of translation-based models. (a) TransE models r as a translation vector. (b) RotatE models r as rotation in complex plane [12]. (c) QuatE using hypercomplex-valued embeddings with three imaginary components to represent entities [13].	10
2.2	Illustrate the message passing process on GAT. Left: A weight vector parameterizes the attention mechanism, using LeakyRelu as the activation function. Right: Multi-head attention performs on a node’s neighbourhood, and each head’s aggregated features are used to obtain the final feature representation of that node.	12
2.3	Illustrate of Retrieval-Augmented QA Task. By calculating the vector similarity to retrieve top K relevant external sources as a prompt, which is later fed into LM together with the question.	14
2.4	Illustrate DiFaR embedding triplets of KG and text to the same representation space to foster directly retrieving the fact in KG corresponding to the text.	16
2.5	Illustrate the integration of KG and LM. Where the question is left untouched, only the graph is encoded by considering the presentation of different layers of the question.	17
3.1	The overall framework. It consists of three different stages: 1. Retrieval; 2. MEDGNP ; 3. Training.	20

3.2	Illustrate of the KG Retrieval. We first get all named entities within the text, comparing them with the concepts in a knowledge base, and then we extract subgraphs based on the connectivity between the pairs of all aligned entities.	21
3.3	We prioritise entities in an order: 1. Red : direct neighbour with option entity; 2. Yellow : within 2-hop neighbour with option entity; 3. Blue : 2-hop neighbour between query entities.	21
3.4	The framework of MEDGNP. It consists of multiple modules that process text and KGs and integrate their information to obtain prompt embedding; we then use this prompt embedding together with the text embedding as input into LMs.	22
3.5	Illustrate of the training process where the LM is frozen while the gradients through the LM are used to train the MEDGNP module.	25
4.1	The LP task dataset creation algorithm.	29
4.2	The GAT design comparison. Left : comparison on number of GAT layers; Right : comparison on relation features strategy.	31
A.3	Illustrate of examples of the datasets used in our experiments.	50
A.4	Illustrate the modified decoder and how to integrate it with the graph embeddings.	51

List of Tables

4.1	The composition details of each QA dataset.	27
4.2	Performance comparison on different models of different datasets. Bold and <u>underline</u> denote the best and second best scores. Δ represent for the improvement between our method and baseline.	30
4.3	Performance comparison of LM Frozen and LM Fine-Tuned. . .	30
4.4	Comparison of different module’s contribution to our model. . .	32
A.5	M is matrix; A is adjacency matrix; D is the corresponding degree Matrix of A; H is the node embeddings at layer l ; W is the weight of layer l ; K is the number of layers.	49
A.6	Illustrate of two settings of query: Setting 1 : hyphen-separated list of options; Setting 2 : prefaces the answers with capitalised letters in parentheses.	50
A.7	Comparison on three sets: using only LM, using LM+MedGNP (Ours) and using LM+deep-integration technique.	51
A.8	Performance comparison on combining different strategies	52

Chapter 1

Theoretical Framework

This chapter briefly introduces the fundamental elements supporting our research, laying the foundation for understanding the theoretical underpinnings of the work presented in this paper.

1.1 Language Models

LMs are fundamental components in the field of NLP, which aims to predict the likelihood of a sequence of words occurring in a given language by learning the patterns and relationships between words. The application scope of LMs has already expanded to encompass a wide range of NLP areas, including machine translation, information retrieval, text summarization, and QA, continually advancing based on new architectures and algorithms [14].

With the rise of deep learning, particularly since the emergence of the transformer architecture in 2017 [15], transformer-based models such as BERT (Bidirectional Encoder Representations from Transformers) [16] and GPT (Generative Pre-trained Transformer) [17] have fundamentally transformed the landscape of NLP research and application. These developments have made it significantly easier for machines to understand and interact with human language at an unprecedented level. Unlike traditional recurrent neural networks or convolutional neural network structures, the attention mechanism is the core of the transformer model, which allows the model to capture and compute any two positions within the sequence directly when processing sequential data, enabling each output to focus flexibly on any part of the input sequence. The essence of attention is a weighted sum. Through an additional neural network layer, it selects certain parts of the input or assigns different weights to different parts of the input.

Figure 1.1 shows that the transformer model comprises of N encoder layers and N decoder layers. The encoder focuses on feature representation, while the

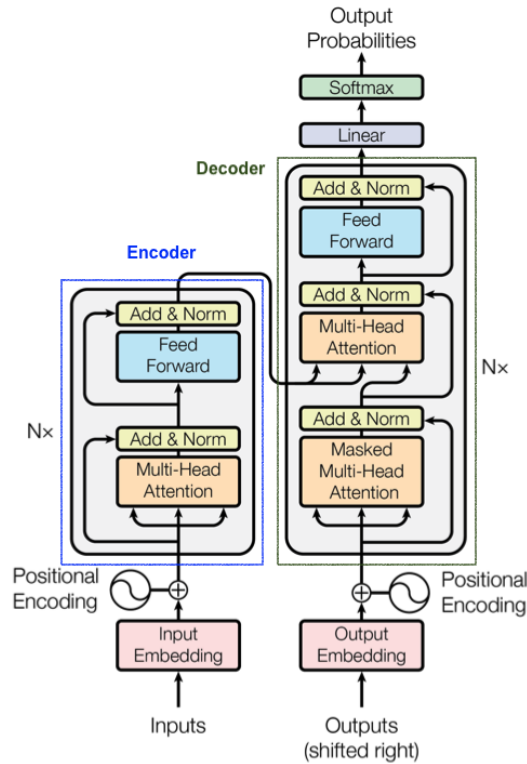


Figure 1.1: The transformer model architecture. It consist of two separate modules: encoder and decoder.

From [15]

decoder excels at text generation. The encoder and decoder comprise several crucial layers that work together to process and generate sequential data. Here are some key layers within the Transformer model:

1. **Self-Attention Layer:** the self-attention layer is the core of the transformer, allowing the model to consider all positions within the sequence when processing each sequence position, thereby capturing the relationships between different parts of the sequence.
2. **Feed-Forward Network (FFN):** positioned after the self-attention layer, the FFN applies the same fully connected layers to each position. It primarily processes the output of the self-attention layer, adding non-linearity on top of it.
3. **Encoder-Decoder Attention Layer (Only in the Decoder):** this layer enables the decoder to focus on the output from the encoder. The decoder uses this layer to concentrate on relevant parts of the input sequence, thereby considering the input's context when generating the sequence.

Combining these layers enables the transformer model to handle complex sequential tasks effectively. The success of the transformer model has also spurred the development of many transformer-based variants [18], rapidly replacing other neural models across different domains and types of NLP tasks due to their powerful capabilities in understanding context and generating text, becoming the new industry standard in NLP.

1.2 Knowledge Graph Partitioning

KG Partitioning involves dividing a large KG into smaller, interconnected segments without compromising the integrity and semantic relationships of the original graph. KGs accurately describe the semantic relationships between various entities and their complex interrelations in the real world. They serve as graph-based data structures across diverse domains [19], from social networks and biomedical sciences to complex industrial systems. As KGs grow in scale and complexity, directly manipulating and querying the entire graph becomes increasingly tricky. Therefore, KG partitioning has emerged as an effective strategy to enhance processing efficiency, query speed, and scalability.

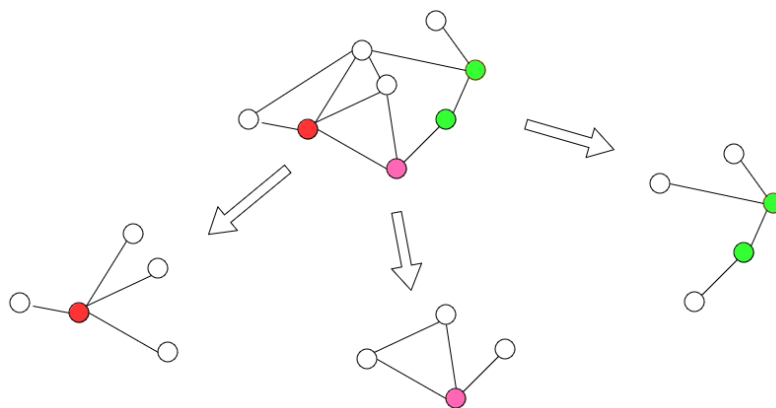


Figure 1.2: Illustrate an example of KG partitioning. Different colours represent different categories of nodes; this partitioning operates based on the types of nodes.

The primary goal of KG partitioning is to distribute the nodes (entities) and edges (relationships) of a global KG into smaller sub-graphs as shown in Figure 1.2. This distribution should minimise the number of inter-partition edges to reduce cross-segment communication and maintain a balance in the size of each partition to ensure even workload distribution [20].

Researchers have developed various partitioning algorithms, including graph-cut based approaches [21], clustering techniques [22], and methods leveraging the semantic structure of KGs [23]. These algorithms typically involve heuristic

or optimisation-based solutions to address the trade-offs between partitioning granularity, interconnectivity, and processing efficiency.

In addition, leveraging a distributed graph database such as NebulaGraph¹ stores graph data across various partitions via hashing and distributing these partitions across all storage nodes, which enables the management of large data volumes with minimal latency. This strategy significantly enhances the speed of graph analytics. Furthermore, NebulaGraph's integration with the llamaindex [24] framework facilitates the swift and direct retrieval of relevant graph partitions. This retrieval is based on KG Partitioning in streamlining the processing and analysis of complex graph-based data.

For scenarios tailored to specific tasks or queries, implementing dynamic partitioning of KGs is key [25]. It allows the system to focus on knowledge relevant to the scenario, dynamically extracting different subsets of knowledge. This flexibility makes KG Partitioning more adaptable to diverse application contexts, enhancing the applicability and practicality of KGs in specific domains.

1.3 Retrieval-Enhanced Model

As an emerging frontier approach in the field of NLP, the retrieval-enhanced model stands out from traditional models that depend solely on pre-trained information embedded during the training phase [26]. By incorporating a mechanism for dynamically accessing external databases or knowledge bases, these models enrich their informational breadth, significantly improving their ability to generate outputs that are accurate, contextually relevant, and rich in detail.

Currently, approaches based on such enhancement mechanisms are attracting increasing interest. For instance, conventional text generation models encounter limitations in open-domain QA systems, especially when the system must adapt to real-time dynamics and integrate the latest data in generating revised answers. Traditional methods store knowledge through extensive parameters and need help to respond to queries outside their pre-loaded information. In contrast, models augmented with external knowledge access can overcome these limitations.

Models like Retrieval Augmented Generation (RAG) [27], featuring components for both searching and generating. The retrieval module fetches pertinent documents, explicitly grounding knowledge. Concurrently, the generation segment, employing an encoder-decoder architecture, integrates the original input with the fetched data, processing them collectively to produce relevant outcomes, this procedure as shown in Figure 1.3.

¹<https://github.com/vesoft-inc/nebula>

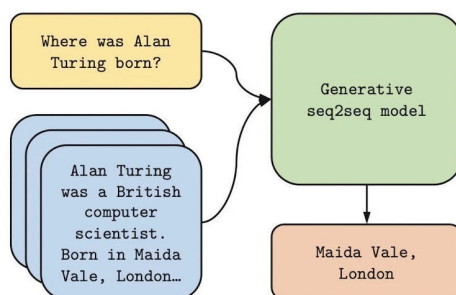


Figure 1.3: Text generation example based on retrieval enhancement.

From [28]

These advanced models present several advantages over their traditional counterparts:

1. Knowledge does not need to be implicitly present in the model’s parameters but can be explicitly introduced plug-and-play, offering greater scalability.
2. Rather than generating text from scratch, the retrieval-enhanced model uses relevant documents through search as a reference, alleviating text generation’s difficulty.

Overall, models through retrieval-enhanced mechanisms exhibit clear advantages in many scenarios. Sole reliance on the vast parameter size of pre-trained LMs could be more satisfactory in domain transfer or knowledge update scenarios. The retrieval-enhanced approach is naturally suited to these scenarios. In some cases, retrieval-enhanced models can achieve performances comparable to large LMs while using much fewer model parameters [29].

1.4 Prompt Tuning

Prompt learning has emerged as a new hotspot in NLP research [30]. Adding a prompt to the input present significantly enhance the performance of the pre-trained model. Both prompt-tuning and fine-tuning are methods for refining pre-trained models. As LMs become increasingly prominent, the cost of fine-tuning them also escalates. Conversely, the prompting method requires no modifications to the pre-trained model but instead involves altering the prompts input into the model. By changing these prompts, the approach effectively transitions the model’s domain from a general scope to a task-specific one [31]. In Figure 1.4 shows prompt-tuning with zero-shot and Figure 1.5 shows one-shot approach.

```
Translate English to French: /* task description */
cheese => __ /* prompt */
```

Figure 1.4: Example of prompt-tuning with zero-shot learning.

```
Translate English to French: /* task description */
sea otter => loutre de mer /* example */
cheese => __ /* prompt */
```

Figure 1.5: Example of prompt-tuning with one-shot learning

Compared to the one-shot approach, as shown in Figure 1.5, The principles of few-shot learning are entirely consistent with this approach, which essentially involves providing multiple examples. Indeed, it has been present that few-shot learning can achieve excellent performance with just 100 examples [31]. Based on the different designs of prompts, we can divide them into two categories:

1. **Hard prompt:** hard prompts refer to manually designed prompts. Typically, hard prompts require the model to have substantial experience in the domain, and users must understand the model’s underlying mechanics before use. Otherwise, the performance of hard prompts tends to be significantly inferior to the state-of-the-art (SOTA) achieved by fine-tuning.
2. **Soft prompt:** Due to the limitations of hard prompts, scientists proposed soft prompts [32]. In contrast to hard prompts, soft prompts treat the generation of prompts as a learnable task, effectively transitioning the creation of prompts from discrete human trial-and-error to continuous machine learning and experimentation.

Prompt-tuning requires the input and output to fit within a template, inevitably necessitating the original task’s format to be restructured to achieve optimal performance. From this, we observe that the essence of prompt tuning is to modify the task format to cater to the performance of large models. In other words, the premise of prompt-tuning is that the performance of the pre-trained model is already excellent, and all we need is to convert the task format at inference time to obtain superior performance.

²<https://zhuanlan.zhihu.com/p/395115779>

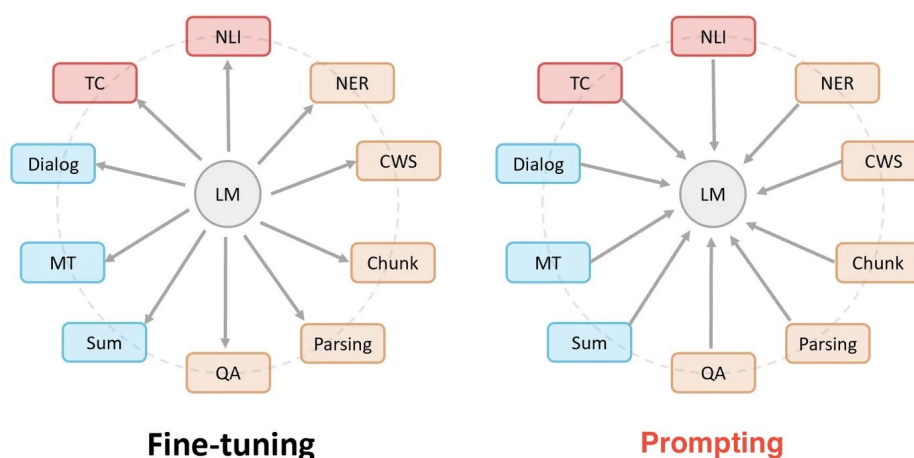


Figure 1.6: Comparison between fine-tuning and prompt-tuning. In the figure, the circles represent the pre-trained LM, and the rectangular boxes represent various downstream NLP tasks.

From (Zhihu)²

Figure 1.6 illustrates the fundamental difference between fine-tuning and prompt-tuning. Both aim to bring pre-trained LMs closer to downstream tasks, but the approach differs. In fine-tuning, the pre-trained LM "adapts" to various downstream tasks. In prompting, various downstream tasks "adapt" to the pre-trained LM.

1.5 UMLS

A medical terminology system is a standardised vocabulary for information exchange, data analysis, research, and decision support. These systems provide a consistent language to describe medical concepts and conditions, allowing healthcare providers, researchers, and decision-makers to communicate and understand medical information accurately and uniformly. Some medical terminology systems organise terms according to concepts, which are then linked to other concepts through various types of relationships [4], thus creating rich graphs and specific hierarchical structures.

UMLS is a comprehensive and integrated compendium from nearly 200 different vocabularies and ontologies, these vocabularies cover various healthcare and biomedical science aspects, ranging from diseases and drugs to medical procedures and equipment, including widely recognised ones such as: SNOMED CT³(Systematised Nomenclature of Medicine – Clinical Terms),

³<https://www.snomed.org/>

MeSH⁴(Medical Subject Headings), and ICD-10⁵(International Classification of Diseases, Tenth Revision). UMLS is developed and maintained by the National Library of Medicine (NLM)⁶. The NLM updates UMLS annually to ensure it reflects the latest knowledge and terminologies in the healthcare and biomedical domains.

It provides a unified framework that enables a semantic network across different domains of medical knowledge, facilitating the integration, retrieval, and analysis of health and biomedical information. The UMLS consists of three main components:

1. ***Metathesaurus***: An extensive, multi-lingual database that contains over 3 million biomedical concepts and 15 million terms sourced from various health and biomedical vocabularies and classifications are interconnected by more than 11 million relationships that represent semantic connections between them. It allows mapping among these concepts and terms across different vocabularies.
2. ***Semantic Network***: Provides a consistent categorisation of all concepts represented in the Metathesaurus, defining the types of relationships that may exist among these concepts.
3. ***SPECIALIST Lexicon and Lexical Tools***: A set of tools and a lexicon designed for natural language processing applications. The lexicon includes syntactic, morphological, and orthographic information for biomedical and general English.

The UMLS is extensively utilised in health informatics applications across tasks such as electronic health record systems, biomedical research, clinical decision support systems, and information retrieval systems. It is available in Rich Release Format (RRF), a set of ASCII text files designed for easy parsing and integration into database management systems. To better leverage UMLS data for analysis and practical applications, the NLM also offers a variety of tools and APIs⁷, such as MetaMap⁸[33]. These tools are designed to support the integration of NLP tasks, such as information retrieval, text mining, categorisation and classification, text summarisation, QA, and knowledge discovery—with the rich concepts and relationships of the UMLS Metathesaurus, thereby enhancing the performance and accuracy of these tasks.

⁴<https://www.nlm.nih.gov/mesh/meshhome.html>

⁵<http://www.who.int/classifications/icd/en/>

⁶<https://www.nlm.nih.gov/>

⁷https://lhncbc.nlm.nih.gov/ii/tools/Terms_of_Service.html/

⁸<https://lhncbc.nlm.nih.gov/ii/tools/MetaMap.html>

Chapter 2

Related Work

Relevant prior work includes studies of Knowledge Graph Embedding (KGE) and knowledge-enhanced LMs. This chapter elucidates and reviews the existing literature on these topics.

2.1 Knowledge Graph Embedding

KGE forms the foundation for the correct and rational application of KGs. Various research studies on KGE inspire our utilisation of the UMLS database.

It is a technique that aims to translate the complex, structured information within a KG as dense vectors or embeddings in a low-dimensional space, where these embeddings capture the semantic similarities and relational patterns among entities [34]. We can widely use the obtained embeddings in a computationally efficient way that benefits various downstream tasks [35] such as prediction, entity category identification, entity disambiguation and other tasks related to graph knowledge components, as well as tasks such as question and answer, recommendation, relationship classification and other tasks related to non-graph knowledge components [36].

Various research methods work to learn how to embed KGs better, and we can roughly divide these methods into three groups: translation-based, embedding-based, and neural network-based models.

Research methods for learning how to get a better embedding of KGs vary widely. These methods can be roughly divided into the following three groups: translation-based, embedding-based and neural network-based models.

2.1.1 Translation-Based Models

The translation-based approach can be understood as treating each triplet in the KG as a translation process from the head entity through the relationship

to the tail entity. The difference between various translation-based models lies in the design of the scoring function (see Appendix 4.3 for details). The classical algorithm is the TransE model [37], which models relationships between entities through simple geometric operations such as vector addition. The RotatE model [12] goes further on the TransE model. It models the relationship r in the triplet (h, r, t) as a rotation of the complex plane, making r more expressive. Going one step further than RotatE, QuatE represents the relationship transformation through a 4-tuple Hamilton Product [13], resulting in a more compact interaction between entities and relations. Translation-based models also include HAKE [38], DualE [39], GIE [40], etc.

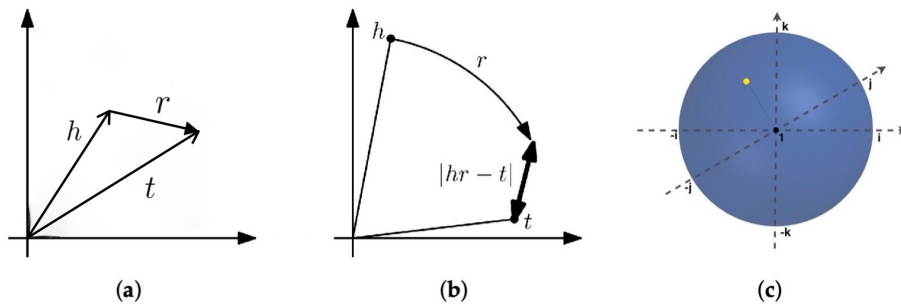


Figure 2.1: Illustrate of translation-based models. (a) TransE models r as a translation vector. (b) RotatE models r as rotation in complex plane [12]. (c) QuatE using hypercomplex-valued embeddings with three imaginary components to represent entities [13].

(b) Taken From [12], (c) Taken From [13]

Such models usually have a simple structure and often possess high computational efficiency [41]. However, they struggle to handle one-to-many, many-to-one, and many-to-many relationships. Therefore, capturing complex relationships in medical domain KGs, such as hierarchical or long-distance dependencies, may require more efficient methods.

2.1.2 Embedding-Based Models

A KG can be considered as a sizeable third-order tensor, where the embeddings of entities and relations can be learned through matrix factorisation. In these models, the likelihood of triplets is typically quantified using the multiplicative operations of embeddings [42]. Typical representatives of this type of model include: RESCAL [43], DistMult [44], ComplEx [45]. Because the scoring functions (see Appendix 4.3 for details) are calculated as bilinear products based on similarity, it is also called a bilinear model.

DistMult [44] simplifies the RESCAL model by constraining the relation matrices to be diagonal. Building on DistMult, ComplEx [45] introduces complex-valued embeddings, where embeddings of entities and relations are no longer confined to a real-valued space but extended to a complex space. Moreover, the scoring function of ComplEx is not symmetric, allowing for different scores for triplets with asymmetric relation types based on the positional relationship of head and tail entities, thereby better modelling asymmetric relations.

While they are generally effective at capturing the semantic relationships between entities and computationally efficient, they also face challenges, such as difficulty in handling asymmetric relations and the potential for overfitting to training data. Due to the diversity and complexity of relation types in medical domain KGs, embedding-based models require appropriate adjustments or combinations with other methods to ensure the models accurately reflect the richness of medical knowledge.

2.1.3 Neural Network-Based Models

Graph data is notoriously complex and challenging to work with. However, with the rapid expansion of deep learning in recent years, researchers have drawn inspiration from convolutional networks, recurrent networks, and deep auto-encoders to define and design neural network structures for handling graph data. The concept of Graph Neural Network (GNN) was first proposed in 2005 [46], and it was submitted to learn the node representation of the graph by iteratively propagating and aggregating neighbour nodes (see Appendix 4.3 for details). Subsequently, Graph Convolutional Network (GCN) extended convolutional operations from traditional data, such as images, to graph data [47]. The GCN model can capture information from K-hop neighbours by stacking multiple GCN layers.

To build on this foundation and address the issue of GNN not accounting for the varying importance of different neighbour nodes during aggregation, Graph Attention Network (GAT) [48] drew inspiration from the idea of transformers, introducing a masked self-attention mechanism as shown in Figure 2.2. In computing the representation of each node in the graph, GAT assigns different weights based on the distinct features of neighbouring nodes.

GNN models have significantly improved the ability to capture rich relationship information in KGs, surpassing traditional embedding technologies in multiple tasks, and GNN models can enrich the representation of KG entities and help solve numerous problems in the NLP field by combining other data modalities, such as text [49]. KGs in the medical field often involve imperfect and heterogeneous data sources. GNN exhibit strong robustness in processing

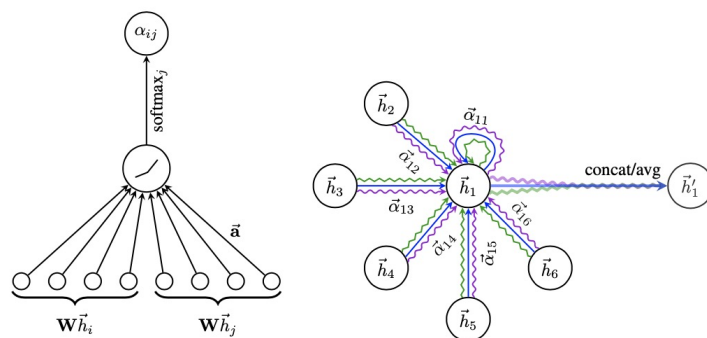


Figure 2.2: Illustrate the message passing process on GAT. **Left:** A weight vector parameterizes the attention mechanism, using *LeakyRelu* as the activation function. **Right:** Multi-head attention performs on a node’s neighbourhood, and each head’s aggregated features are used to obtain the final feature representation of that node.

From [48]

such incomplete and noisy data [50], making them particularly suitable for processing data in the medical field. However, GNN models usually assume that graphs are static. However, graphs in the real world may be dynamic or incomplete, especially in the medical field, where new research discoveries may lead to adding new entities and relationships or updating existing knowledge. Therefore, designing GNN models suitable for specific medical applications still needs to consider the specificity of the data and how to integrate and utilise the available medical knowledge effectively.

2.1.4 Integration with Our Work

The GAT model [48] is most relevant to our work. Although the distance model provides an intuitive and computationally efficient method for KG representation [12, 13, 37, 38, 39, 40], considering the complex relationships between entities [4] in medical KGs, more sophisticated models may be required to capture better and utilise the complexity and richness of medical KGs.

In addition to the complexity of relationships within medical KGs, the standardisation of medical terminologies poses a significant challenge to achieving optimal entity embeddings. CODER [51] employs terms and relation triplets from the KG to address the term standardisation issue. It calculates the semantic similarity between terms and incorporates term-relation-term similarity to obtain superior medical embeddings.

We adopt the form of a GAT combined with a LM. First, by using a LM that is trained on a large-scale biomedical literature library [52], we can fully capture the rich semantic information when obtaining the entity embedding

of the medical KG, and then during the GAT training process, the entity embeddings are considered alongside the structural information in the graph, enabling the model to learn both the semantic features of entities and the structural features of the graph.

In addition, we design a self-supervised Link Prediction (LP) task to represent KG further [53] through a DistMult model. We used triples formed by random sampling and constructing negative samples as input to the DistMult model to circumvent the limitations of semantic matching models in handling asymmetric relationships. Our work aims to achieve balance and better representation of the KG by combining the strengths of different models and complementing each other’s limitations.

2.2 Knowledge-Enhanced Language Models

LMs can learn knowledge from large-scale corpora and achieve state-of-the-art performance in various NLP tasks such as entity recognition and relation extraction [14]. However, when generating text, LMs may produce plausible but incorrect content, a phenomenon known as the hallucination problem [54]. Furthermore, although LMs can learn vast knowledge from large corpora, this knowledge is based on statistical patterns rather than genuine understanding, potentially leading to a lack of depth and accuracy when addressing specific domain issues. Lastly, the outputs generated by LMs often need more interpretability [55], making it easier for users to understand how the model arrived at a particular conclusion or produced specific text. This lack of interpretability limits the model’s usability in critical tasks.

To solve these problems, knowledge grounding LMs has been a focal point of active research [56]. Retrieval-augmented LMs have emerged as a promising line of work, focusing on grounding additional knowledge from the relevant text from a corpus to LMs [57, 58, 59], Figure 2.3 shows the use of this approach to solve the QA task. These models have presented significant performance gains across diverse corpora, underscoring the utility of integrating retrieval mechanisms into LMs [58]. Additionally, the development of models like Atlas [60], with a focus on learning knowledge-intensive tasks with very few training examples, highlights the potential of retrieval-augmented LMs in leveraging limited data for effective learning [57].

In parallel, the relevance of using KGs as background knowledge to ground reasoning about entities and facts in LMs has also garnered attention. With the advent of transformer architectures [15], LMs have demonstrated unparalleled versatility and adaptability across diverse contexts and applications, becoming a significant milestone in the domain of NLP. At the same time, KG [61]

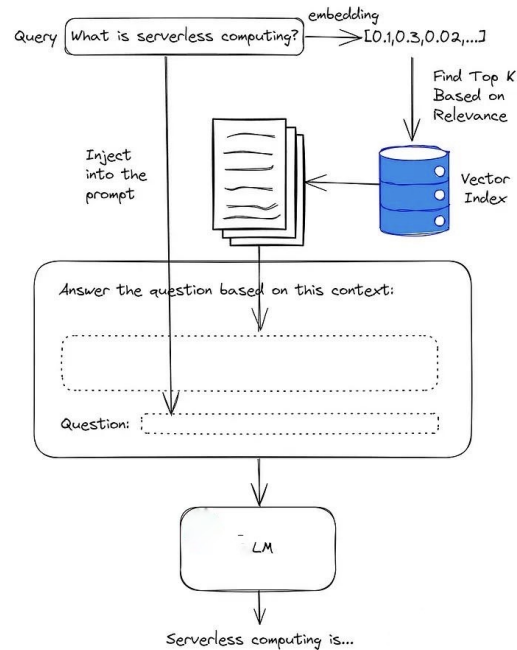


Figure 2.3: Illustrate of Retrieval-Augmented QA Task. By calculating the vector similarity to retrieve top K relevant external sources as a prompt, which is later fed into LM together with the question.

embodies cutting-edge knowledge engineering in artificial intelligence. LMs primarily encode knowledge implicitly within their parameters, which can lead to challenges such as the fabrication of facts and a lack of interpretability. In contrast, KG represents structured, explicit knowledge. They inherently possess high interpretability and deliver high-quality, domain-specific knowledge. By leveraging KGs to provide additional, structured, and high-quality knowledge, it is possible to mitigate some of the inherent limitations of LMs, such as their propensity for factual inaccuracies and their often opaque decision-making processes. Integrating KGs with LMs represents a promising avenue towards developing more reliable, interpretable, and contextually informed AI systems [62].

Early attempts include adding entities of KGs as additional input features to LMs such as K-BERT [63], ERNIE 3.0 [64] and CoLAKE [65]. Subsequent research focused more on effectively integrating and utilising the structural information of KGs, such as encoding entities and relationships in KG through graph neural networks [62, 66].

In summary, the current realm of kg-enhanced LMs research predominantly bifurcates into two principal methodologies: retrieval-augmented approaches [60, 67] and deep integration techniques [11, 68, 69]. Each strategy leverages

the strengths of KG to enhance the performance and applicability of LMs.

2.2.1 Retrieval-Augmented Approaches

Retrieval-augmented strategies harness the predictive capabilities of LMs by enriching them with a vast reservoir of structured knowledge. This method entails extracting pertinent information from a sizeable KG during the generation or inference process to assist in reasoning [29, 58, 70, 71]. Its primary advantage lies in the dynamic utilisation of extensive external knowledge, significantly broadening the model’s reference frame and potentially leading to more accurate and contextually relevant outcomes. However, this approach may require additional computational resources to manage the retrieval and integration process effectively [72].

Transforming the KG into a series of key-value pair embeddings is a promising strategy to improve efficiency during retrieval processes and minimise the computational burden [73, 74]. This approach enables the utilisation of Maximum Inner Product Search (MIPS), implemented via libraries such as Faiss [75] for rapidly retrieving entity or complete triple information from the KG.

Models such as QA-GNN [76] exemplify this approach by identifying relevant information within large KGs to capture the nuances of question contexts. It improves the efficiency and accuracy of responses by scoring the relevance of KG nodes and conducting joint reasoning. Despite its advantages, QA-GNN may inadvertently introduce entities unrelated to the question’s context, potentially diluting the quality of the inference.

Furthermore, JointLK [77] represents a novel attempt to address the challenge of noisy nodes within subgraphs and the limited interaction between language representations and KG representations. It facilitates multi-step joint reasoning between the LM and the KG, enabling a deeper exchange of information across modalities. This capability supports a more nuanced understanding and interpretation of complex queries, bridging the gap between textual and structured knowledge. Nevertheless, the practical application of JointLK may demand substantial computational resources, posing challenges for scalability and efficiency.

Unlike the aforementioned approach, which starts retrieval from nodes, DiFaR [78], based on representational similarities, directly retrieves facts from the KG as shown in Figure 2.4. This approach simplifies the complex scenarios encountered in traditional retrieval that require multiple tasks, such as entity span detection, entity disambiguation, and relation classification, making knowledge acquisition more efficient. At the same time, KELM [72] uses these triplets to generate synthetic natural language sentences to maintain

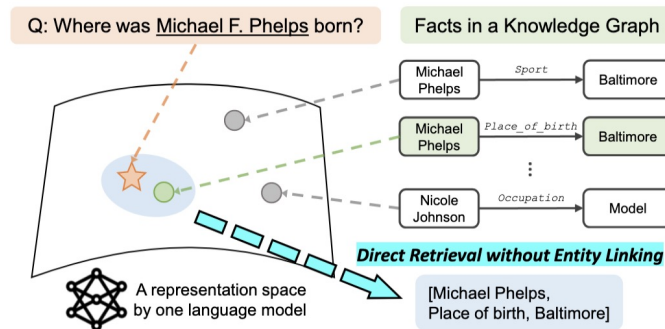


Figure 2.4: Illustrate DiFaR embedding triplets of KG and text to the same representation space to foster directly retrieving the fact in KG corresponding to the text.

Taken From [78]

consistency in the format between triplets and text for better integration of information from multiple triplets without altering the model architecture.

In addressing the efficiency of integrating retrieved knowledge into model training, recent advancements such as the use of RETRO [29] and prompt-based methods [79] have significantly reduced the number of parameters required for model training. Akin to the same research interest of ours, BioReader [80] adopts this innovative approach for biomedical NLP and exemplifies the application of these methods in the medical field, highlighting their potential to enhance the performance of models tasked with interpreting complex medical queries by knowledge grounding.

In essence, while retrieval-augmented methods offer a promising avenue for leveraging KGs to augment LMs, they also introduce considerations regarding computational overhead and the relevance of retrieved information. Particularly in our task where we focus on medical QA, methods that convert KG into textual formats or attempt to directly amalgamate retrieved KG information with context capitalise on the comprehension capabilities of LMs. However, these approaches risk losing some of the inherent structural information present within KG. Such relationships are crucial for capturing the depth and complexity of medical knowledge, from the hierarchical organisation of diseases and symptoms to the multifaceted interactions between various treatments and patient conditions.

2.2.2 Deep Integration Techniques

Following the retrieval-augmented approaches, deep integration techniques represent a sophisticated methodology that embeds KG information directly within the architectural LMs. This method facilitates a profound knowledge grounding LM, enabling a synergistic enhancement of both elements. The

primary advantage of deep integration lies in its capacity for comprehensive fusion and interaction across the entire architectural spectrum of the model, fostering a seamless blend of textual and structured knowledge.

However, this integration depth comes with challenges, notably requiring substantial computational resources and intricate model designs to manage the complexity of effectively merging these two distinct information sources. GreaseLM [81] stands as a pioneering model in this realm, striving for a genuinely unified amalgamation of KG and LM. It champions a holistic approach to integration, permitting nuanced interactions and coalescence at all model architecture levels.

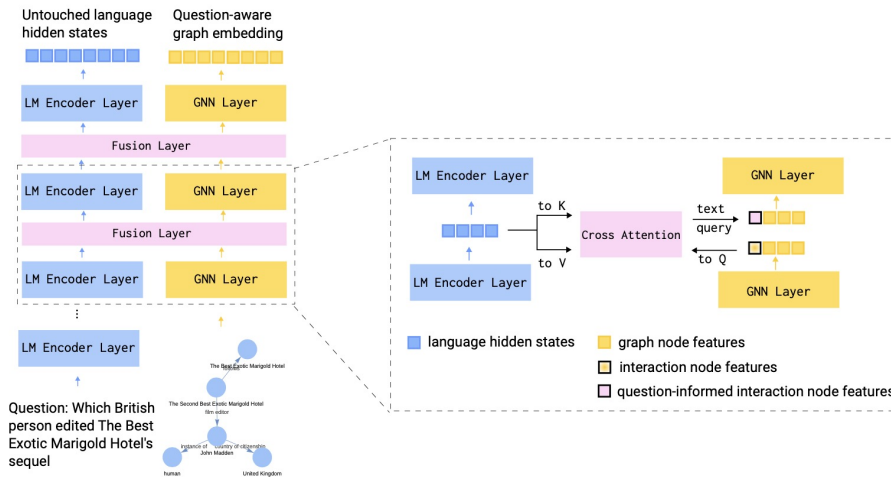


Figure 2.5: Illustrate the integration of KG and LM. Where the question is left untouched, only the graph is encoded by considering the presentation of different layers of the question.

From [68]

Conversely, GraphextQA [68] offers an alternative approach by adopting a frozen LM strategy within an encoder-decoder transformer architecture. This method first adopts an integration technique close to that of GreaseLM’s to merge the graph’s context with the language’s hidden states during the encoding phase, as shown in Figure 2.5, allowing the LM to guide the graph encoder. Then, in the decoding phase, the language’s hidden states attentively interact with the graph’s hidden states, re-injecting knowledge from the graph back into the language. This bifurcated fusion strategy enables a dynamic exchange of information between the graph and language domains, potentially offering a more efficient pathway to grounding knowledge into LMs.

2.2.3 Integration with Our Work

To address the complexities and computational demands associated with deep integration techniques [68, 81], as well as the potential loss of structural information inherent in retrieval-augmented approaches [29, 58, 70, 71, 73, 74, 80]. In our work, we have embraced a unique approach inspired by GNP’s promising use of text-aware graph embedding as a prompt for knowledge grounding. Unlike GNP, our MEDGNP method explicitly targets the QA task within the medical domain. To better handle the severe imbalance between question entities and answer entities encountered in medical QA datasets, we use a design different from GNP called prioritised neighbour retrieval to maximise the effectiveness of extracted subgraphs from medical texts. Additionally, the medical domain’s KGs, such as UMLS, often feature complex structures. These complexities can hinder the convergence of node embeddings during iterative training. To tackle this challenge, MEDGNP adopts a graph encoding method different from GNP, enhancing our model’s ability to navigate and leverage the intricacies of medical KGs.

By integrating these strategic enhancements, MEDGNP aims to harness the interpretive power of LMs while maintaining the integrity and utility of graph-structured knowledge, thereby facilitating more accurate and contextually informed responses to medical queries.

Chapter 3

Method

This chapter will elaborate on how we employ our method to get knowledge grounded LMs and introduce the design and specific implementation of each module within our proposed MEDGNP method.

3.1 Overview

We explore a methodological approach to tackle the complexities of QA tasks within the medical domain. Pre-trained models such as BERT and GPT have showcased remarkable capabilities in various QA domains [14, 82]. They typically input the model with tokenized textual queries X to predict and generate the corresponding answer y . Our experiment transcends this traditional approach by incorporating additional prompt tokens P derived from KGs. These tokens are amalgamated with the query text tokens and fed into the model, making this task look like:

$$y = f(X) \quad \rightarrow \quad y = f([P, X])$$

thereby endowing the model to leverage structured, grounded knowledge for generating more precise and informative answers.

These prompt tokens are learnable vectors. Unlike task-specific prompts [32, 31], the prompt tokens in our experiment are sample-specific [83]. We design a unique prompt for each query to elicit the highest probability of reflecting the corresponding grounded knowledge encapsulated in the query.

The workflow of our task is structured into three main segments as shown in Figure 3.1: KG retrieval, knowledge grounding via MEDGNP, and model training.

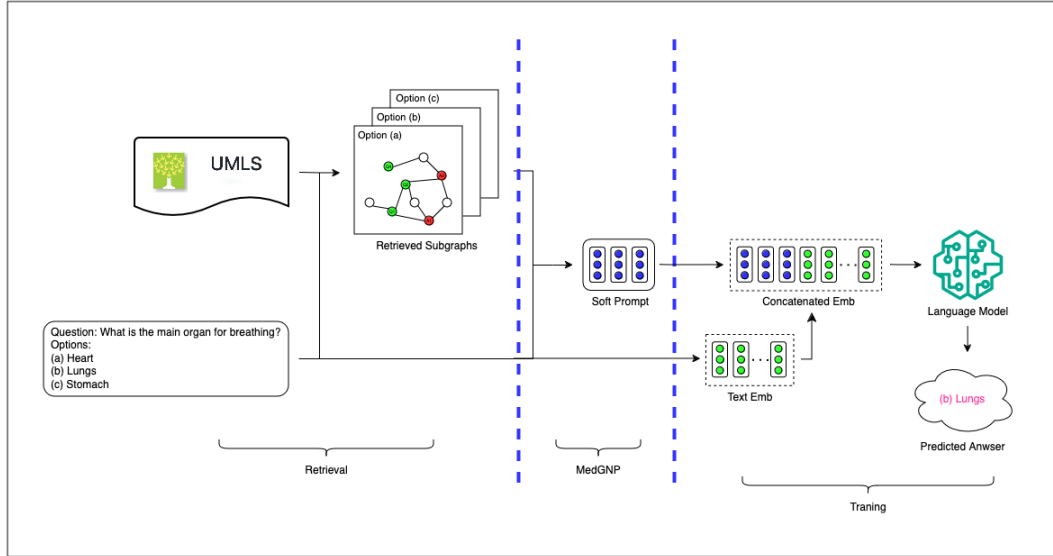


Figure 3.1: The overall framework. It consists of three different stages: 1. Retrieval; 2. MEDGNP ; 3. Training.

3.2 Subgraph Retrieval

Our objective is to efficiently retrieve subgraphs relevant to the queries from the expansive UMLS, which comprises millions of interconnected nodes (see Section 1.5 for details). As shown in Figure 3.2, the retrieval process begins with entity linking, where we first detect the named entities in the text, and then the entities within the text are aligned with their counterparts in the KG. This alignment is a foundation for constructing subgraphs, which we achieve by including aligned entities, their two-hop neighbours, and the relational pathways that interconnect them [76].

For free-form multiple-choice datasets [84], we tailor our retrieval approach to encompass subgraphs corresponding to the combination of different options a_k and their related questions. We must prune these subgraphs to a uniform size to facilitate parallel processing. Given that medical QA datasets typically offer a more extended context text field [85], which can be combined with the question text field to form the query, the entities aligned with the KG from this query far outnumber those from the options. This results in a substantial inclusion of two-hop neighbours related to the entities within the query in the subgraphs.

In light of this, we eschew using random sampling for pruning subgraphs [86]. Instead, we implement a prioritised neighbour approach as shown in Figure 3.3: we begin by adding entities corresponding to the options, followed by incorporating directly related query entities and their two-hop neighbour

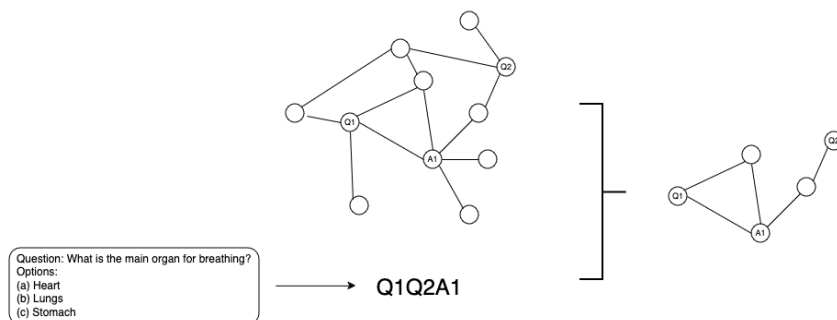


Figure 3.2: Illustrate of the KG Retrieval. We first get all named entities within the text, comparing them with the concepts in a knowledge base, and then we extract subgraphs based on the connectivity between the pairs of all aligned entities.

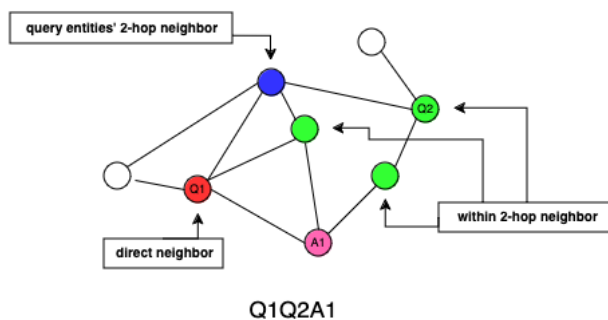


Figure 3.3: We prioritise entities in an order: **1. Red:** direct neighbour with option entity; **2. Yellow:** within 2-hop neighbour with option entity; **3. Blue:** 2-hop neighbour between query entities.

entities starting from the option entities. Finally, we complete the subgraphs by appending all query entities and their respective two-hop neighbour entities. This strategy ensures retaining the most pertinent information related to the options, enhancing the subgraphs' relevance and efficacy for the subsequent processing stages.

3.3 MEDGNP

MEDGNP represents the linchpin in our approach to integrating KGs into LMs. The MEDGNP framework, as illustrated in Figure 3.4, serves as a blueprint for the methodology we adopt. In the following sections, we will delineate the implementation of each step in the workflow, closely following the sequence outlined in the framework diagram.

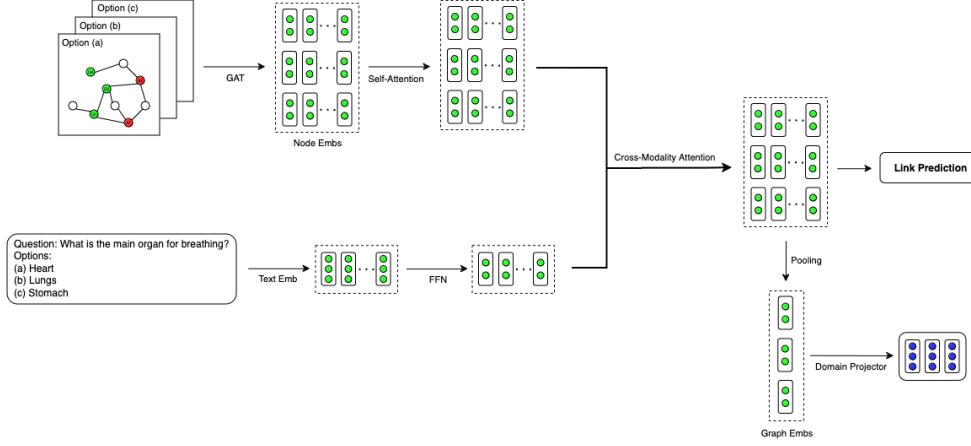


Figure 3.4: The framework of MEDGNP. It consists of multiple modules that process text and KGs and integrate their information to obtain prompt embedding; we then use this prompt embedding together with the text embedding as input into LMs.

3.3.1 Graph Encoder

We commence by leveraging a LM pre-trained on medical domain data [51, 86, 87] to derive initial embeddings $H_{\text{init}} = (n_1, n_2, \dots, n_n)$ for each node within the retrieved subgraph \mathcal{G} . Subsequently, we employ a GAT to enhance the representational capacity of these embeddings concerning the graph’s structural attributes.

Building on the standard GAT framework [48], we incorporate a linear layer to transform the initial embeddings, establishing initial residual connections [88] with the output at every layer of the GAT. This approach ensures efficiency and avoids the potential convergence of node representations within the same connectivity domain to identical values as the number of GAT layers and training iterations increases [89].

$$H_1 = f_{\text{GAT-encoder}}(H_{\text{init}}) \quad (3.1)$$

The procedure updates node representations, yielding H_1 (Equation 3.1), which embody both the semantic richness imparted by the LM and the topological context encoded by the GAT.

3.3.2 Self-Attention Module

We managed to address the limited capacity of GNN models in capturing long-range dependencies between nodes by further refining the node embeddings

obtained from the previous step. We accomplish this by incorporating a self-attention layer that employs a self-attention mechanism to dynamically allocate attention weights to every pair of nodes within the graph, which gives us H_2 (Equation 3.2) as node embeddings obtained after calculating self-attention. This approach helps leverage global graph structure information to discern the relative importance of nodes within the graph structure.

$$H_2 = f_{\text{Self-Attention}}(H_1) \quad (3.2)$$

3.3.3 Text Embedding and Transformation

To facilitate the interaction between the nodes in the graph and the latent semantic states of the textual queries, thereby enabling the flow of information from language to graph, it is essential to generate text embeddings aligned with the node embeddings' dimensions. The medical QA datasets are typically rich in information [85], necessitating an effective integration strategy for model input.

For this purpose, we start by assimilating the diverse information into a standardised pattern following some template [90] (see Appendix 4.3), thereby crafting a single-sentence text. Following this, first, we acquire the text embedding \mathcal{T} through the embedding layer of a LM. Subsequently, we employ a FFN to transform \mathcal{T} , aligning it with the dimensionality of the node embeddings, resulting in an updated text embedding \mathcal{T}' (Equation 3.3).

$$\mathcal{T}' = f_{\text{FFN}_1}(\mathcal{T}) \quad (3.3)$$

Harmonising the linguistic and graph-based representations is a crucial step, setting the stage for the subsequent fusion and interaction within our model framework.

3.3.4 Cross-Modality Attention Module

Leveraging the node embeddings H_2 derived in Section 3.3.2 as queries, and the text embeddings \mathcal{T}' obtained in Section 3.3.3 serving as both keys and values, we facilitate an intricate exchange of information across data modalities [91]. A cross-modality attention module accomplishes this by discerning the nodes within the graph most pertinent to the text.

$$H_3 = f_{\text{Cross-Modality Attention}}(H_2, \mathcal{T}') \quad (3.4)$$

The fusion of modalities synergistically harnesses the salient features of text and graph, providing an enhanced representation of node embeddings H_3 (Equation 3.4).

3.3.5 Pooling and Domain Projector

By applying average pooling to consolidate all node embeddings within a retrieved subgraph \mathcal{G} , we acquire a singular vector G (Equation 3.5). It is a graph-level representation [92] which considered the node significance in the structure of \mathcal{G} and the correlation with the text.

To merge this graph-level embedding with textual embeddings, thereby equipping the LM with the ability to exploit graph-structured information, we implement another FFN. This FFN operation adjusts the dimensions of the graph embedding, aligning it with the text embedding dimensions to enable seamless integration into the LM.

$$\begin{aligned} G &= f_{\text{Pooling}}(H_3) \\ \mathcal{M} &= f_{\text{FFN}_2}(G) \end{aligned} \quad (3.5)$$

The culmination of the pooling and domain projector operations results in forming the final prompt vector \mathcal{M} (Equation 3.5).

3.3.6 Self-Supervised Link Prediction

Furthermore, we dive into an additional LP task to validate the comprehensive learning of the KG structure using our node embeddings. Capitalising on the inputs derived from Section 3.3.4 where node embeddings are refined via cross-attention mechanisms to incorporate textual information, this task can also be seen as encourage the model to predict missing linkages within the graph by synergizing the structural properties of the KG with the corresponding textual content.

To operationalize this, we randomly sample a subset of triples $\langle h, r, t \rangle$ from the graph, where h and t is the head and tail entity and r is the relation between them, which is later treated as positive instances for our training set. To generate negative samples, we substitute the head or tail entities with other nodes from the graph that do not share a connection with the given triple. Then we utilise DistMult [44] as our scoring function: $\phi_r(h, t) = \langle h \cdot r \cdot t \rangle$, which effectively measures the plausibility of a triple by considering the multiplicative interactions between the entities and relations. The training objective is to optimize the combined score of positive S_{pos} and negative S_{neg} samples in the formula 3.6:

$$\mathcal{L}_{\text{LP}} = \sum_{(h,r,t) \in S_{\text{pos}}} \left(-\log \sigma(\phi_r(h, t) + \gamma) + \frac{1}{n} \sum_{(h',r,t') \in S_{\text{neg}}} \log \sigma(\phi_r(h', t') + \gamma) \right) \quad (3.6)$$

where $-\log \sigma(\phi_r(h, t) + \gamma)$ denotes the scores of the positive triples. In contrast, $\frac{1}{n} \sum_{(h',r,t') \in S_{\text{neg}}} \log \sigma(\phi_r(h', t') + \gamma)$ denotes for the scores of the negative triples,

γ represents the margin: a hyperparameter that delineates the buffer between positive and negative samples and σ is the sigmoid function.

3.4 Training with Language Model

The training process for our MEDGNP, as depicted in Figure 3.5, adopts a novel approach by freezing the parameters of the pre-trained LM. This methodology, which we refer to as directly prompting the LM, involves training external modules while maintaining the LM in a static state. This strategy aligns with or exceeds the performance benchmarks set by traditional fine-tuning methods while necessitating minimal or no updates to the model parameters [93].

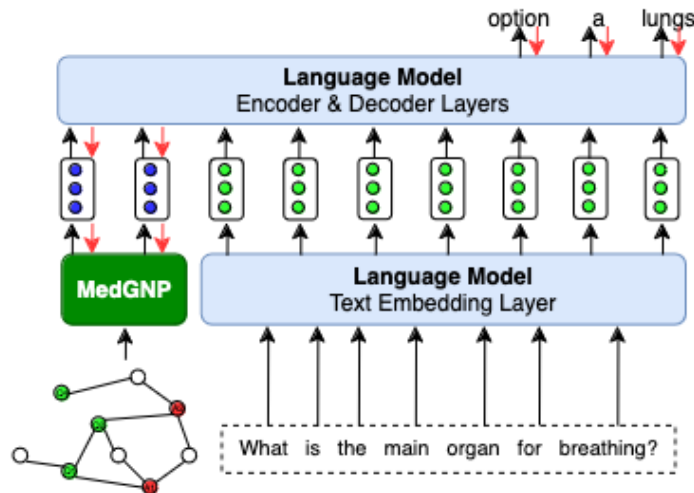


Figure 3.5: Illustrate of the training process where the LM is frozen while the gradients through the LM are used to train the MEDGNP module.

To train MEDGNP with the LM, we compute the cross-entropy between the ground truth and the model predictions, which serves as the loss function for the LM. This loss function is combined with the link prediction loss discussed in Section 3.3.6, resulting in our final objective function (Equation 3.7) for training.

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda \cdot \mathcal{L}_{\text{LP}} \quad (3.7)$$

Here, λ represents the weight parameter applied to the link prediction loss.

Incorporating continuous prompts at the beginning of a sequence and fine-tuning the parameters associated with these prompts enables us to guide and control the model's behaviour effectively.

Chapter 4

Experiments and Results

This chapter delves into the comprehensive experimental framework and the resultant insights derived from applying MEDGNP to medical domain QA tasks. Through meticulous experimental design, results analysis, and ablation studies, we aim to elucidate our proposed method’s efficacy and intricate workings, providing a thorough validation of its performance against established benchmarks.

4.1 Experimental Setup

4.1.1 Data Preparation

We conduct experiments on the medical domain QA task. For the KG data, we use two selected sources of the UMLS [4] most commonly support medical QA datasets, and together, they contain about 300K nodes and 1.2M edges. **Diseases Database, 2000**¹ is a cross-referenced index of human disease, medications, symptoms, signs, and abnormal investigation findings. The content focuses on internal medicine, inherited disease, clinical biochemistry, and pharmacology.

NCI Thesaurus, 2022_09D² provides definitions, synonyms, and other information on cancers and related diseases, single agents and related substances, and other topics related to cancer and biomedical research.

We use three multiple-choice QA datasets to evaluate MedGNP; examples can be found in Appendix 4.3, and composition details are shown in Table 4.1. We set the maximum tokens of query from the QA dataset to 512, and when pruning the retrieved subgraph as discussed in Section 3.2, we set the maximum of nodes to 200 for each subgraph.

¹<http://www.diseasesdatabase.com/>

²<http://ncit.nci.nih.gov/>

PubMedQA [94] is a 3-way multiple-choice dataset built from PubMed articles that use binary questions as titles. The conclusive parts of the abstracts are the long answers, while the main task of PubMedQA is to predict their short forms (e.g., yes/no/maybe), using the abstracts without the conclusive parts as contexts.

BioASQ [95] is a 2-way multiple-choice dataset built from manually annotated biomedical semantic indexing and QA. BioASQ is a challenge match that provides a rich dataset containing various questions. These questions and answers are extracted from various biomedical literature, including PubMed articles. The Task 7b dataset of BioASQ is used in our experiments.

MedQA-USMLE (MedQA) [84] is a 4-way multiple-choice dataset which is collected from the professional medical board exams covering three languages: English, simplified Chinese, and traditional Chinese. We used the English subset in our experiments.

	PubMedQA	BioASQ	MedQA
# Train	800	672	10184
# Val	104	80	1272
# Test	104	144	1280

Table 4.1: The composition details of each QA dataset.

4.1.2 Model configuration

Accuracy is used as the evaluation metric. We choose *google/flan-t5-base*³ [96] as the LM for training, we set the learning rate to 1e-4, training epochs to 50 and batch size to 8. we use RAdam [97] as the optimiser and linear warmup with ratio 0.1 to schedule the learning rate. For the GAT discussed in Section 3.3.1, we set 1024 as the hidden dimension and two as the attention heads. All training sessions were conducted on SLURM servers provided by UNIBO, utilising NVIDIA GeForce RTX 3090 GPUs with 24GB of RAM.

4.1.3 Implementation Details

This section aims to elucidate the technical underpinnings and the algorithmic strategies we have employed to tackle the challenges presented in our research.

The first algorithm, prioritised neighbour retrieval, as discussed in Section 3.2, is a foundational element in our data processing pipeline for handling KGs.

³<https://huggingface.co/google/flan-t5-base>

This procedure is shown in Algorithm 1, where the \mathcal{Q} and \mathcal{A} is a dictionary containing question and answer’s concept as the key and the 1-hop neighbours to that concept as the value. \mathcal{N} is our desired prioritised neighbour nodes, which will be used to build the subgraph; c_1 and c_2 Represent the priority node and the second priority node, respectively. We start the search from the answer concept, considering that, usually, in medical QA datasets, the concepts from the answers are far less than those from the query. Our algorithm traverses them to detect overlaps in their 1-hop neighbours. This process is pivotal to establishing a potential 2-hop connection between the nodes; if our answer concept is successfully 1-hop or 2-hop connected with the query concept, we save all the extra nodes between them as the priority node and the query node as the second priority node. Last, by combing them all, we get a subgraph that retrieves the answer concept connectivity prioritises.

Algorithm 1: The prioritised neighbour retrieval algorithm.

input : \mathcal{Q} : query entity and it’s 1-hop neighbours.
 \mathcal{A} : answer entity and it’s 1-hop neighbours.
output : \mathcal{N} : subgraph nodes
 $c_1 \leftarrow set()$;
 $c_2 \leftarrow set()$;
for $a \in \mathcal{A}$ **do**
 for $q \in \mathcal{Q}$ **do**
 $connect \leftarrow set(\mathcal{A}[a]) \cap set(\mathcal{Q}[q])$;
 if $connect$ **then**
 $c_2.add(q)$;
 $c_1 \models connect$;
 $\mathcal{N} \leftarrow set(a_1, a_2, a_3, \dots, a_n) \cup (set(c_1), set(c_2))$;
Output: \mathcal{N}

To construct a dataset for the LP task as talked in Section 3.3.6, we commence by extracting positive sample triples. Our entire graph data is transformed into graph data by using library `torch_geometric`⁴, which facilitates the rapid extraction of these triples. The `H.edge_index` represents the graph connectivity in COO format with the shape `[2, num_edges]`, where the two rows of the two-dimensional array correspond to the source (head) and target (tail) nodes of the edges, respectively. A proportion parameter is set to control the volume of positive sample data required. Subsequently, a filtering mask is employed to obtain the corresponding random positive samples.

⁴<https://pytorch-geometric.readthedocs.io/en/latest/index.html>

Once we obtain the corresponding positive sample dataset, we can efficiently generate negative samples for training our LP task by selecting random node pairs and replacing either the head or tail node of the positive samples. The training of the LP task aids our MedGNP model in further learning the structural information of the graph.

```

#all graphs save as torch_geometric graph data,
E = len(H.edge_index[0])
positions = torch.arange(E)      ---- get relations index
                                   ---- H.edge_index is edge_index stored in the shape [2, num_edges].

drop_count = int(len(positions) * drop_probability)
drop_idxxs = torch.multinomial(torch.full((len(positions),), 1.0), drop_count, replacement=False)
drop_positions = positions[drop_idxxs]

                                   ---- we selection a proportion from the whole KG.
                                   ---- Randomly select 'drop_count' non-repeating indexes.

mask = torch.zeros((E,)).long()
mask = mask.index_fill_(dim=0, index=drop_positions, value=1).bool()

                                   ---- transfer the mask tensor to bool.
                                   ---- for later selection our postive samples.

pos_edge_index = H.edge_index[:, mask]
pos_edge_type = H.edge_lp[mask]
pos_triples = [pos_edge_index[0], pos_edge_type, pos_edge_index[1]]

                                   ---- Applying mask, we get our targeted positive triplets.

```

Figure 4.1: The LP task dataset creation algorithm.

4.2 Results

4.2.1 Performance Comparison

To ensure that the proposed method effectively infuses KG information into a LM, we benchmarked against a standalone LM as a baseline. Furthermore, we compared our approach with other state-of-the-art models, including DRAGON [86], PubmedBERT [98], BioBERT [52]. The performance results are presented in Table 4.2. As shown in the table, the integration of prompt-based input significantly enhances the LM’s performance across various datasets. With the implementation of MEDGNP on Flan-T5-base, we observed an improvement of 6.0% on the PubMedQA dataset, 4.29% on BioASQ, and 1.26% on MedQA. These results underscore the efficacy of our approach in leveraging structured medical knowledge to augment the predictive capabilities of LMs.

Model	# training params	PubMedQA	BioASQ	MedQA
DRAGON	360M	73.4	96.4	47.5
PubmedBERT	110M	55.8	<u>87.5</u>	<u>38.1</u>
BioBERT	110M	60.2	84.1	36.7
Flan-T5-base	-	56.0	77.14	26.73
+ MedGNP(Ours)	30M	<u>62.0</u>	81.43	27.99
Δ	-	6.0% \uparrow	4.29% \uparrow	1.26% \uparrow

Table 4.2: Performance comparison on different models of different datasets. **Bold** and underline denote the best and second best scores. Δ represent for the improvement between our method and baseline.

Comparing all the models, DRAGON achieves the best performance across all models but uses the most training parameters, 12X of ours. PubmedBERT comes next. Our model performs comparably to PubmedBERT and BioBERT on the BioASQ dataset. On PubMedQA, our model outperforms them while using 3.6X fewer parameters. It should be noted that both our baseline and our models are generalist models, not pre-training on medical domain datasets, while the comparing models are all pre-trained on large medical domain corpus.

Even in the face of such a knowledge-intensive challenge that requires a comprehensive grasp of medical knowledge and the ability to understand complex terminologies, our method helps general models without such knowledge and capabilities. It enables them to achieve performance comparable to specialised models like BioBERT and PubmedBERT with 3.6X fewer parameters and DRAGON with 12X fewer parameters. Notably, our model even outperforms BioBERT and PubmedBERT on the PubMedQA dataset. However, the performance of our model on MedQA does not measure up to these models, suggesting that further optimisation and design improvements might be needed.

	#training params	LM-only	LM+MedGNP	Improve
Frozen	30M	77.14	81.43	\uparrow 4.29%
Fine-Tune	278M	85.71	87.86	\uparrow 2.15%

Table 4.3: Performance comparison of LM Frozen and LM Fine-Tuned.

To rigorously assess the efficacy of MEDGNP on grounding knowledge into LMs and help them solve knowledge-intensive challenges, we conducted further experiments utilising the BioASQ dataset for fine-tuning the model. This experiment aims to evaluate whether MEDGNP maintains its effectiveness even when

the model satisfies the task requirements with high accuracy. From Table 4.3, it is evident that although the improvement achieved by employing MEDGNP in a fine-tuned setting is not as substantial as in the frozen scenario (from 4.29% to 2.15%), it still significantly aids the LM, presents MEDGNP’s capacity to enhance model performance by effectively incorporating structured medical knowledge, even when the underlying LM has been optimised for the task. Such findings underscore the adaptability of MEDGNP, highlighting its value as a versatile tool for augmenting LMs across various training configurations.

Notably, by keeping the LM parameters frozen, our training involved only 30M parameters — merely about 10% of the entire model’s parameters. This strategic constraint underscores the efficiency of MEDGNP and highlights its resource-saving advantage.

4.2.2 GAT Design Comparison

In each iteration, each node obtains messages from nodes in its one-hop neighbourhood to compute its updated embedding. Therefore, when unrolled for each node, the layers from the output end indicate that the layers effectively pass messages from multiple hop neighbourhoods based precisely on how many layers. In some cases, these distant layer neighbours may be of little help, or they may indeed be helpful in efficiently aggregating the required information. For this purpose, we search the number of GAT layers from $[2, 3, 4, 5]$.

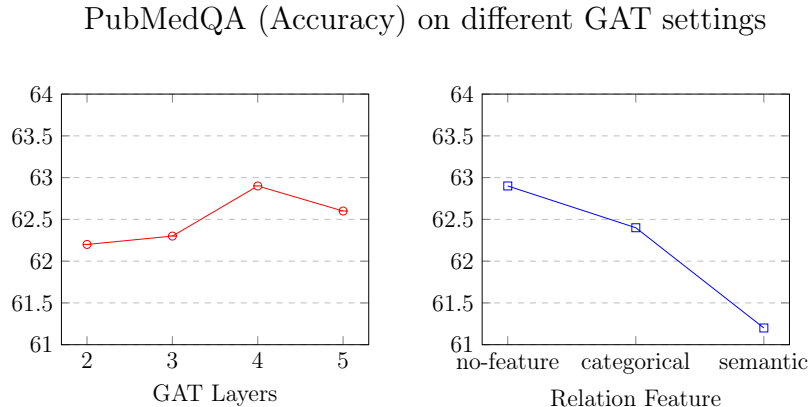


Figure 4.2: The GAT design comparison. **Left:** comparison on number of GAT layers; **Right:** comparison on relation features strategy.

Recent research efforts in GNNs have shifted from simple undirected graphs toward accounting for the diversity of relation types and characteristics within multi-relational graphs. [99, 100]. To address this complexity, we explore three distinct approaches: one disregarding relational features, another utilising

encoded identifiers as a surrogate for relation types, and a third leveraging the semantic textual features of the relations. Each strategy is designed to probe how relational information can be integrated and its impact on the model’s performance. By comparing these approaches, we aim to discern the significance of relation-specific features in enhancing the model’s ability to reason and learn from complex graph structures. [101].

As illustrated in Figure 4.2, the number of layers in the GAT has a discernible impact on the model’s performance. It is observed that the model achieves the highest accuracy with a four-layer GAT configuration. Notably, the outcomes solely based on node information and graph structural insights surpass those where relation features are utilised.

4.3 Ablation Study

To evaluate the efficacy of each module constituting our MEDGNP framework, we conducted a series of experiments wherein each module was individually omitted, enabling us to dissect and understand its unique contribution to the model’s collective performance.

	PubMedQA	BioASQ	MedQA
<i>w/o</i> CMA	60.0	79.93	25.11
<i>w/o</i> SA	61.0	80.33	27.01
<i>w/o</i> DP	60.0	78.33	24.21
<i>w/o</i> LP	61.0	78.83	27.57
MedGNP	62.0	81.43	27.99

Table 4.4: Comparison of different module’s contribution to our model.

As shown in Table 4.4, where CMA (cross-modality attention module), SA (self-attention module), DP (domain projector module: needs to set dimension of GAT match with dimension of LMs) and LP (link prediction module). In conclusion, our MEDGNP framework consistently outperforms across the board, a testament to the formidable efficacy of its composite elements.

Within our MEDGNP framework, the domain projector module emerges as particularly impactful, demonstrating greater significance than other modules in enhancing the model’s performance. This distinction likely stems from the domain projector’s substantial share of the model’s parameter count, second only to the GAT module. This phenomenon is supported by existing research, which indicates that the efficacy of soft prompts improves with a larger parameter space [31].

In the case of the MedQA dataset, the cross-modality attention module also plays a pivotal role. Notably, for the MedQA dataset, in scenarios where either the cross-modality attention module or the domain projector module is omitted, the model’s performance not only fails to improve but diminishes when compared to our baseline model, which solely utilises a LM.

Furthermore, to substantiate the efficacy of our model compared to other methods, we explored and experimented with designs divergent from our own. Specifically, we scrutinised DRAGON [86], a model employing deep integration techniques (2.2.2). Unlike our approach, DRAGON initially processes each option alongside the question separately as a binary classification task, effectively distinguishing true/false statements among multiple choices to identify the correct answer. To ensure a fair comparison, we adapted DRAGON to a sequence-to-sequence (seq2seq) model akin to our MEDGNP method, utilising the flan-T5 framework, the detail implementation and results can be found in Appendix 4.3.

Conclusions and Future Challenges

We presented MEDGNP, a novel approach for infusing structured medical knowledge into LMs to enhance performance in medical QA tasks. By leveraging advanced techniques such as GAT, cross-modal attention modules, and self-supervised learning, MEDGNP present significant improvements by grounding knowledge efficiently to LMs. This achievement underscores the importance of incorporating structured knowledge into LMs, highlighting a promising direction for future research in NLP and medical informatics.

Despite the success of MEDGNP, some limitations remain to be addressed. For instance, the range of KGs used in the model could be expanded to include a more comprehensive array of medical knowledge. This change could potentially increase the model’s efficacy and applicability to a broader range of medical questions, and the design of multiple soft prompts still needs to be considered in terms of how to inform the model correctly, especially in a situation with multiple KGs given at the same time.

Furthermore, future studies should explore the applicability of the MEDGNP framework in other fields, such as legal or financial domains. This exploration will demonstrate the versatility of the approach and provide insights into the universal benefits of integrating structured knowledge into LMs.

Acknowledgments

At this moment, I express my gratitude for my parents' silent dedication and tender care. Their behind-the-scenes effort has allowed me to complete this journey of life steadfastly.

I am thankful for the patient academic support and kind personal care from all professors and tutors. Thanks to my supervisor, Gianluca, for his guidance and leading me to delve deeper into this field. I also thank my co-supervisor Giacomo for his constant support throughout the process of completing my thesis.

I am grateful to all my friends, whether they are in China, Italy, or perhaps from some unknown corner of the universe. They have always been there for me, helping and caring for me and I wish them all the best.

Looking back on everything, it feels like a dream. I hope that upon waking, I will still be that carefree person.

曾怡然

8 March 2024

Bibliography

- [1] Saskia Locke, Anthony Bashall, Sarah Al-Adely, John Moore, Anthony Wilson, and Gareth B. Kitchen. Natural language processing in medicine: A review. *Trends in Anaesthesia and Critical Care*, 38:4–9, 2021.
- [2] Da Yin, Li Dong, Hao Cheng, Xiaodong Liu, Kai-Wei Chang, Furu Wei, and Jianfeng Gao. A survey of knowledge-intensive nlp with pre-trained language models, 2022.
- [3] Pietro Di Lena, Giacomo Domeniconi, Luciano Margara, and Gianluca Moro. Gota: Go term annotation of biomedical literature. *BMC Bioinformatics*, 16, 2015.
- [4] Olivier Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(Database issue):D267–D270, 2004.
- [5] Tian Kang, Adler Perotte, Youlan Tang, Casey Ta, and Chunhua Weng. UMLS-based data augmentation for natural language processing of clinical research literature. *Journal of the American Medical Informatics Association*, 28(4):812–823, 12 2020.
- [6] Besnik Fetahu, Zhiyu Chen, Sudipta Kar, Oleg Rokhlenko, and Shervin Malmasi. Multiconer v2: a large multilingual dataset for fine-grained and noisy named entity recognition, 2023.
- [7] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. Gpt-ner: Named entity recognition via large language models, 2023.
- [8] Thilini Wijesiriwardene, Vinh Nguyen, Goonmeet Bajaj, Hong Yung Yip, Vishesh Javangula, Yuqing Mao, Kin Wah Fung, Srinivasan Parthasarathy, Amit P. Sheth, and Olivier Bodenreider. Ubert: A novel language model for synonymy prediction at scale in the umls metathesaurus, 2022.

-
- [9] V Nguyen and O Bodenreider. Adding an attention layer improves the performance of a neural network architecture for synonymy prediction in the umls metathesaurus. *Stud Health Technol Inform*, 290:116–119, 2022.
- [10] Bastien Liétard, Mikaela Keller, and Pascal Denis. A tale of two laws of semantic change: Predicting synonym changes with distributional semantic models, 2023.
- [11] Yijun Tian, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, Nitesh V. Chawla, and Panpan Xu. Graph neural prompting with large language models, 2023.
- [12] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space, 2019.
- [13] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. Quaternion knowledge graph embeddings, 2019.
- [14] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.
- [15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.
- [16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
- [17] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.
- [18] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers:

- State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.
- [19] 王鑫, 陈蔚雪, 杨雅君, 张小旺, and 冯志勇. 知识图谱划分算法研究综述. *计算机学报*, 44(1):235–260, 2021.
- [20] Zainab Abbas, Vasiliki Kalavri, Paris Carbone, and Vladimir Vlassov. Streaming graph partitioning: an experimental study. *Proc. VLDB Endow.*, 11(11):1590–1603, jul 2018.
- [21] Dawei Zhou, Si Zhang, Mehmet Yigit Yildirim, Scott Alcorn, Hanghang Tong, Hasan Davulcu, and Jingrui He. A local algorithm for structure-preserving graph cut. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 655–664, New York, NY, USA, 2017. Association for Computing Machinery.
- [22] Sharnil Pandya and Swarndeep Saket. An overview of partitioning algorithms in clustering techniques. *International Journal of Electrical and Computer Engineering*, 5, 09 2020.
- [23] Kegong Shi, Jinjin Yan, and Jinquan Yang. A semantic partition algorithm based on improved k-means clustering for large-scale indoor areas. *ISPRS International Journal of Geo-Information*, 13(2), 2024.
- [24] Jerry Liu. LlamaIndex, 11 2022.
- [25] Hyeryun Park, Jiye Son, Jeongwon Min, and Jinwook Choi. Selective umls knowledge infusion for biomedical question answering. *Scientific Reports*, 13, 08 2023.
- [26] Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. A survey on retrieval-augmented text generation, 2022.
- [27] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.
- [28] Konstantinos Andriopoulos and Johan Pouwelse. Augmenting llms with knowledge: A survey on hallucination prevention, 2023.

-
- [29] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens, 2022.
- [30] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021.
- [31] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021.
- [32] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation, 2021.
- [33] Alan R Aronson. Metamap: Mapping text to the umls metathesaurus. *Bethesda, MD: NLM, NIH, DHHS*, 1:26, 2006.
- [34] Xiou Ge, Yun-Cheng Wang, Bin Wang, and C. C. Jay Kuo. Knowledge graph embedding: An overview, 2023.
- [35] Yankai Lin, Xu Han, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Knowledge representation learning: A quantitative review, 2018.
- [36] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.
- [37] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
- [38] Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. Learning hierarchy-aware knowledge graph embeddings for link prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):3065–3072, Apr. 2020.

-
- [39] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Dual quaternion knowledge graph embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):6894–6902, May 2021.
- [40] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Geometry interaction knowledge graph embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5):5521–5529, Jun. 2022.
- [41] 区恩海. 基于组合关系翻译的知识表示学习模型. *计算机科学与应用*, 12(3):654–661, 2022.
- [42] Iliara Ferrari, Giacomo Frisoni, Paolo Italiani, Gianluca Moro, and Claudio Sartori. Comprehensive analysis of knowledge graph embedding techniques benchmarked on link prediction. *Electronics*, 11(23), 2022.
- [43] Maximilian Nickel, Volker Tresp, and Peer Kröger. A three-way model for collective learning on multi-relational data. pages 809–816, 01 2011.
- [44] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases, 2015.
- [45] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction, 2016.
- [46] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 2:729–734 vol. 2, 2005.
- [47] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.
- [48] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.
- [49] Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. Graph neural networks for natural language processing: A survey, 2022.
- [50] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications, 2021.

-
- [51] Zheng Yuan, Zhengyun Zhao, Haixia Sun, Jiao Li, Fei Wang, and Sheng Yu. Coder: Knowledge infused cross-lingual medical term embedding for term normalization, 2021.
- [52] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, September 2019.
- [53] Nicolas Hubert, Pierre Monnin, Armelle Brun, and Davy Monticolo. Knowledge graph embeddings for link prediction: Beware of semantics! In *DL4KG@ ISWC 2022: Workshop on Deep Learning for Knowledge Graphs, held as part of ISWC 2022: the 21st International Semantic Web Conference*, 2022.
- [54] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren’s song in the ai ocean: A survey on hallucination in large language models, 2023.
- [55] Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. Rethinking interpretability in the era of large language models, 2024.
- [56] Karthik Soman, Peter W Rose, John H Morris, Rabia E Akbas, Brett Smith, Braian Peetoom, Catalina Villouta-Reyes, Gabriel Cerono, Yongmei Shi, Angela Rizk-Jackson, Sharat Israni, Charlotte A. Nelson, Sui Huang, and Sergio Baranzini. Biomedical knowledge graph-enhanced prompt generation for large language models. *ArXiv*, abs/2311.17330, 2023.
- [57] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane A. Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Few-shot learning with retrieval augmented language models. *ArXiv*, abs/2208.03299, 2022.
- [58] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.
- [59] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. *ArXiv*, abs/2310.01558, 2023.

-
- [60] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models, 2022.
- [61] Amit Singhal. Introducing the knowledge graph: things, not strings, 2012. Accessed on 2020-11-13.
- [62] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. Greaselm: Graph reasoning enhanced language models for question answering. *ArXiv*, abs/2201.08860, 2022.
- [63] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2901–2908, Apr. 2020.
- [64] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiayang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation, 2021.
- [65] Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. Colake: Contextualized language and knowledge embedding, 2020.
- [66] Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. Knowledge graph-augmented language models for knowledge-grounded dialogue generation, 2023.
- [67] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In Hal Daurm e III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR, 13–18 Jul 2020.
- [68] Yuanchun Shen, Ruotong Liao, Zhen Han, Yunpu Ma, and Volker Tresp. Graphextqa: A benchmark for evaluating graph-enhanced large language models, 2023.

-
- [69] Yuequn Wang, Liyan Dong, Hao Zhang, Xintao Ma, Yongli Li, and Minghui Sun. An enhanced multi-modal recommendation based on alternate training with knowledge graph representation. *IEEE Access*, 8:213012–213026, 2020.
- [70] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training, 2020.
- [71] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.
- [72] Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training, 2021.
- [73] Yuxiang Wu, Yu Zhao, Baotian Hu, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. An efficient memory-augmented transformer for knowledge-intensive nlp tasks, 2022.
- [74] Jinheon Baek, Alham Fikri Aji, Jens Lehmann, and Sung Ju Hwang. Direct fact retrieval from knowledge graphs without entity linking, 2023.
- [75] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.
- [76] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering, 2022.
- [77] Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering, 2022.
- [78] Jinheon Baek, Alham Fikri Aji, Jens Lehmann, and Sung Ju Hwang. Direct fact retrieval from knowledge graphs without entity linking. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10038–10055, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [79] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases?, 2019.

- [80] Giacomo Frisoni, Miki Mizutani, Gianluca Moro, and Lorenzo Valgimigli. BioReader: a retrieval-enhanced text-to-text transformer for biomedical literature. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5770–5793, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.
- [81] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. Greaselm: Graph reasoning enhanced language models. In *International Conference on Learning Representations*, 2021.
- [82] Santanu Bhattacharjee, Rejwanul Haque, Gideon Maillette de Buy Weninger, and Andy Way. Investigating query expansion and coreference resolution in question answering on bert. In Elisabeth Métais, Farid Meziane, Helmut Horacek, and Philipp Cimiano, editors, *Natural Language Processing and Information Systems*, pages 47–59, Cham, 2020. Springer International Publishing.
- [83] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models, 2021.
- [84] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams, 2020.
- [85] Qiao Jin, Zheng Yuan, Guangzhi Xiong, Qianlan Yu, Huaiyuan Ying, Chuanqi Tan, Mosha Chen, Songfang Huang, Xiaozhong Liu, and Sheng Yu. Biomedical question answering: A survey of approaches and challenges. *ACM Computing Surveys*, 55(2):1–36, January 2022.
- [86] Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D Manning, Percy Liang, and Jure Leskovec. Deep bidirectional language-knowledge graph pretraining, 2022.
- [87] Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. Scalable multi-hop relational reasoning for knowledge-aware question answering, 2020.
- [88] Jingbo Zhou, Yixuan Du, Ruqiong Zhang, and Rui Zhang. Adaptive depth graph attention networks, 2023.
- [89] Xiang Song, Runjie Ma, Jiahang Li, Muhan Zhang, and David Paul Wipf. Network in graph neural network, 2021.

-
- [90] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. The flan collection: Designing data and methods for effective instruction tuning, 2023.
- [91] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion, 2022.
- [92] Guixiang Ma, Nesreen K. Ahmed, Theodore L. Willke, and Philip S. Yu. Deep graph similarity learning: A survey, 2020.
- [93] Niall Taylor, Yi Zhang, Dan W. Joyce, Ziming Gao, Andrey Kormilitzin, and Alejo Nevado-Holgado. Clinical prompt learning with frozen language models. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–11, 2023.
- [94] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [95] George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael Alvers, Dirk Weißenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artieres, Axel-Cyrille Ngonga Ngomo, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, and Georgios Paliouras. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16:138, 04 2015.
- [96] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

-
- [97] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*, April 2020.
- [98] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23, October 2021.
- [99] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks, 2017.
- [100] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks, 2020.
- [101] Yijun Tian, Kaiwen Dong, Chunhui Zhang, Chuxu Zhang, and Nitesh V. Chawla. Heterogeneous graph masked autoencoders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9997–10005, Jun. 2023.
- [102] Giacomo Frisoni, Paolo Italiani, Stefano Salvatori, and Gianluca Moro. Cogito ergo summ: Abstractive summarization of biomedical papers via semantic parsing graphs and consistency rewards. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):12781–12789, Jun. 2023.
- [103] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022.

Appendix

Knowledge Graph Representation Models

Table A.5 summarise the specific embedding dimensions and space complexity of the different types of models discussed in Section 2.1.

Types	Model	Entity Embedding	Relation Embedding	Score Function
Translation based	TransE	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$-\ \mathbf{h} + \mathbf{r} - \mathbf{t}\ _{1/2}$
	RotatE	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{r} \in \mathbb{C}^d$	$\ \mathbf{h} \circ \mathbf{r} - \mathbf{t}\ $
	QuatE	$\mathbf{h}, \mathbf{t} \in \mathbb{H}^d$	$\mathbf{r} \in \mathbb{H}^d$	$\mathbf{h} \otimes \frac{\mathbf{r}}{ \mathbf{r} } \cdot \mathbf{t}$
Embedding based	RESCAL	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{M}_r \in \mathbb{R}^{d \times d}$	$\mathbf{h}^T \mathbf{M}_r \mathbf{t}$
	DistMult	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\mathbf{h}^T \text{diag}(\mathbf{M}_r) \mathbf{t}$
	ComplEx	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{r} \in \mathbb{C}^d$	$\text{Re}(\langle \mathbf{h}, \mathbf{r}, \bar{\mathbf{t}} \rangle)$
Neural Network based	GCN	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\sigma\left(\mathbf{D}^{-\frac{1}{2}} \bar{\mathbf{A}} \mathbf{D}^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{w}^{(l)}\right)$
	GAT	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\sigma\left(\frac{1}{K} \sum_{k=1}^K \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{w}^k \hat{h}_j\right)$

Table A.5: M is matrix; A is adjacency matrix; D is the corresponding degree Matrix of A ; H is the node embeddings at layer l ; W is the weight of layer l ; K is the number of layers.

Datasets Example and Preprocessing Pattern

We provide an overview illustrations of the sample dataset employed in our experiments in Figure A.3. Furthermore, as delineated in Section 3.3.3, we elucidate the rationale behind selecting preprocessing patterns for each sample within the datasets.

In our endeavour to streamline disparate datasets into a cohesive structure, we adopted a methodology that translates these diverse sources into a unified format. We take into consideration two different template settings [90]; one generates a hyphen-separated list of options, and the other prefaces the answers

MedQA-USMLE	<p>A 57-year-old man presents to his primary care physician with a 2-month history of right upper and lower extremity weakness. He noticed the weakness when he started falling far more frequently while running errands. Since then, he has had increasing difficulty with walking and lifting objects.</p> <p>His past medical history is significant only for well-controlled hypertension, but he says that some members of his family have had musculoskeletal problems. His right upper extremity shows forearm atrophy and depressed reflexes while his right lower extremity is hypertonic with a positive Babinski sign. Which of the following is most likely associated with the cause of this patients symptoms?</p> <p>(A) HLA-B8 haplotype (B) HLA-DR2 haplotype (C) Mutation in SOD1 (D) Mutation in SMN1</p>
PubMedQA	<p>Recent studies have demonstrated that statins have pleiotropic effects, including anti-inflammatory effects and atrial fibrillation (AF) preventive effects [...]</p> <p>221 patients underwent CABG in our hospital from 2004 to 2007. 14 patients with preoperative AF and 4 patients with concomitant valve surgery [...]</p> <p>The overall incidence of postoperative AF was 26%. Postoperative AF was significantly lower in the Statin group compared with the Non-statin group (16% versus 33%, p=0.005). Multivariate analysis demonstrated that independent predictors of AF [...]</p> <p>Do preoperative statins reduce atrial fibrillation after coronary artery bypass grafting?</p> <p>(A) yes (B) no (C) maybe</p>
BioASQ	<p>LT4 absorption is unchanged by concomitant metformin ingestion. It has been hypothesized that metformin may suppress serum thyrotropin (TSH) concentrations by enhancing LT4 absorption or by directly affecting the hypothalamic-pituitary axis. Does metformin interfere thyroxine absorption?</p> <p>(A) yes (B) no</p>

Figure A.3: Illustrate of examples of the datasets used in our experiments.

From [86]

with capitalised letters in parentheses as shown in Table A.6. This facilitates the maintenance of consistency across the datasets and ensures the model's responses are balanced and accurate.

In our experimentation, the first setup (as "Setting 1" shown in Table A.6) yielded better results than the other one. However, the choice of these templates needs to be tailored to the specific circumstances of each dataset.

sample: question*: Which organ in the human body is responsible for filtering and removing waste and excess water from the blood?; options*: Heart, Kidney, Liver, Pancreas.	
Setting 1:	Setting 2:
<p>Question: Which organ in the human body is responsible for filtering and removing waste and excess water from the blood?</p> <p>Options: -Heart. -Kidney. -Liver. -Pancreas.</p>	<p>Question: Which organ in the human body is responsible for filtering and removing waste and excess water from the blood?</p> <p>Options: (A) Heart. (B) Kidney. (C) Liver. (D) Pancreas.</p>

Table A.6: Illustrate of two settings of query: **Setting 1**: hyphen-separated list of options; **Setting 2**: prefaces the answers with capitalised letters in parentheses.

Alternative Method

This adaptation involved contextualising graph nodes with language-hidden states during the encoding phase, shown in Figure 2.5. Additionally, we

modified decoder blocks to incorporate gated-cross-attention mechanisms. In scenarios involving multiple options, which correspond to various subgraphs, we introduce an FFN following the application of gated-cross-attention to each subgraph to combine their outcomes effectively [102], the architecture shown in Figure A.4. This adjustment allows for the seamless integration of knowledge into the language decoding process, thereby aligning DRAGON’s operational paradigm with the seq2seq model structure employed by MEDGNP. But the training params significantly increased because of the deep integration

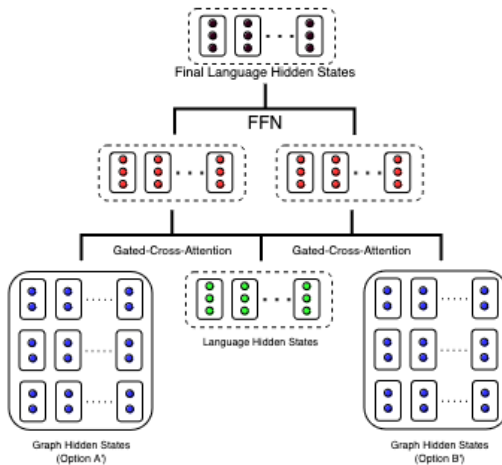


Figure A.4: Illustrate the modified decoder and how to integrate it with the graph embeddings.

with every block of the LM, even while keeping the LM frozen. To mitigate the potential adverse effects of randomly initialised graph encoders and cross-attentions at early stages, we follow the approach of Flamingo [103]. We employ a gating mechanism to ensure that the performance of the original LM remains unaffected at initialisation.

Model	#params	PubMedQA	BioASQ	MedQA
Flan-T5-base	248M	56.0	77.14	26.73
LM + MedGNP(Ours)	278M	62.0	81.43	27.99
LM + <i>deep-integration</i>	418M	<u>57.0</u>	<u>77.14</u>	<u>27.20</u>

Table A.7: Comparison on three sets: using only LM, using LM+MedGNP (Ours) and using LM+deep-integration technique.

We get the result in table A.7, where we can see that the using of deep integration techniques does help LM, yet when compared to our MEDGNP

approach, our method achieves superior results across all datasets. Notably, MEDGNP operates with only 30M training parameters, significantly fewer than the 170M parameters required by deep integration techniques. Our MEDGNP method conserves training resources and outperforms deep integration techniques by achieving exceptional performance with merely 17% of the parameters needed for deep integration. This efficiency underscores MEDGNP’s effectiveness in leveraging a leaner parameter set to deliver enhanced outcomes, presenting its superiority in optimising resource utilisation while maintaining, if not exceeding, the quality of results.

PubMedQA on different model strategies

	LM-only	LM+MedGNP	LM+ <i>deep-integration</i>
MedGNP+ <i>deep-integration</i> 59.0	56.0	62.0	57.0
	↑ 3%	↓ -3%	↑ 2%

Table A.8: Performance comparison on combining different strategies

We also put an experiment to explore combining our MEDGNP approach with methods based on deep integration. The results provide insightful observations on the compatibility and effectiveness of leveraging both strategies within our model framework.

The findings from this experiment shown in Table A.8 are intriguing, deep integration methods predicated on node-level knowledge infusion, stand in contrast to our MEDGNP approach that leverages graph-level knowledge. Initially, we hypothesised that a hybrid model combining graph information at different levels would enhance the model’s performance by utilising a multi-granularity knowledge base.

However, the accuracy achieved with this combined approach was 59%, a modest 3% improvement over the baseline model that solely utilises the LM, which scored 56%. While this combined method did surpass the performance of deep integration techniques alone, which cut 57%, it fell short of the 62% accuracy attained by our standalone MEDGNP method.

This outcome suggests that while integrating knowledge at different levels of granularity can yield performance gains over a baseline LM, the most effective strategy in our case remains the graph-level knowledge infusion provided by MEDGNP. This could imply that the graph-level approach of MEDGNP more efficiently captures and utilises the structural and relational complexities inherent in the KG, translating to superior performance over combining it with node-level infusions.

