Fine-tuning NER for Triplet Extraction in Medical Knowledge Graph Construction

Richard Reinhart* & Masayu Leylia Khodra

School of Electrical and Informatics Engineering, Institut Teknologi Bandung,
Jalan Ganesa No. 10 Bandung 40132, Indonesia
*E-mail: richard.rein16@gmail.com

Abstract. This study presents a new approach for constructing a medical knowledge graph using Named Entity Recognition (NER) to identify entities such as diseases, drugs, or medical procedures, alongside part-of-speech (POS) tagging and dependency parsing to determine words that function as verbs and roots. These extracted words are then used as relations between entities, forming triplets in the format (entity, relation, entity). While the knowledge graph provides a structured representation of medical information, the evaluation primarily reflects the performance of the underlying NLP pipeline (NER, POS tagging, and dependency parsing) used to generate the triplets. Quantitative evaluation was performed using metrics such as precision, recall, and F1-score to assess the accuracy and completeness of entity and relation extraction. The qualitative evaluation involved medical domain experts to assess the relevance and validity of the relationships derived. The results indicate that fine-tuning a pre-trained model for NER and leveraging a pre-trained model for POS tagging and dependency parsing can effectively generate accurate triplets for constructing a medical knowledge graph. This approach demonstrated strong performance, achieving high evaluation scores in both quantitative and qualitative evaluations.

Keywords: *dependency parsing; medical knowledge graph; named entity recognition; part-of-speech tagging; triplets.*

1 Introduction

The rapid expansion of medical knowledge, coupled with the increasing volume of clinical data, necessitates the development of more efficient methods for extracting, organizing, and utilizing healthcare information. Among the vast array of data sources, clinical notes—written by healthcare professionals during patient interactions—are a particularly rich and underutilized resource. These notes contain valuable information regarding patient conditions, diagnoses, treatments, and outcomes, but their unstructured nature makes it challenging to extract meaningful insights.

Knowledge graphs (KGs) have emerged as a powerful tool for structuring and representing complex information in graph format, where entities (such as diseases, medications, and symptoms) are nodes, and relationships (such as

Received January 12th, 2025, Revised October 9th, 2025, Accepted for publication 6 November 2025.

‘causes,’ ‘treats,’ or ‘associated_with’) form the edges [1]. By leveraging knowledge graphs, clinical data can be systematically organized, enabling better decision-making by doctor and personalized healthcare delivery [2]. However, constructing high-quality knowledge graphs from clinical texts remains an ongoing challenge due to the inherent complexities of natural language, including ambiguity, varied terminology, and the nuanced relationships between entities.

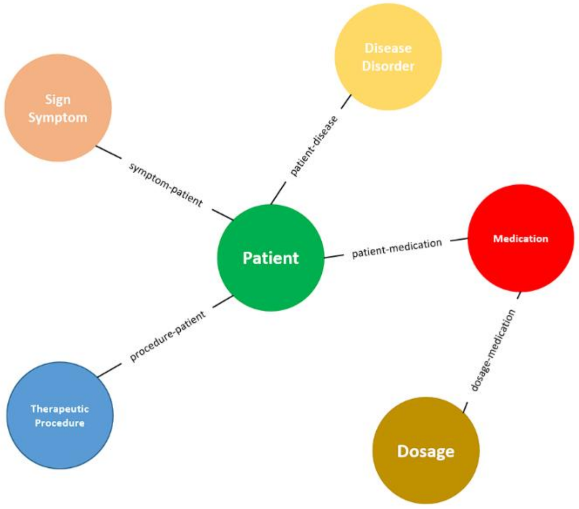

Figure 1 KGs with simple edges.

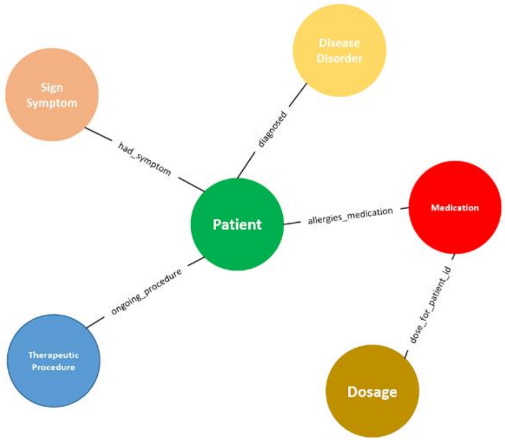

Figure 2 KGs with contextual edges.

In this work, we propose an approach to enhance the construction of knowledge graphs from medical clinical notes by using advanced natural language processing (NLP) techniques. Specifically, we focus on three key NLP tasks, Named entity recognition (NER), dependency parsing, and part-of-speech (POS) tagging. NER is used to identify and classify entities (e.g., diseases, drugs, symptoms) [3], while dependency parsing and POS tagging are employed to extract the grammatical structure of sentences and identify the root verbs that define relationships between entities [4]. The core contribution of this paper is the use of these linguistic features to generate more informative and contextually rich edges in the knowledge graph, thereby capturing the complex interactions between medical concepts with greater precision [5].

Current methods for knowledge graph construction often rely on simple co-occurrence-based edges or predefined relationship types, which can overlook the subtleties of how medical concepts interact in the text. By incorporating syntactic analysis through dependency parsing and POS tagging, we aim to capture richer, context-dependent relationships that better reflect the clinical narrative. For instance, rather than simply linking 'drug X' and 'disease Y' as co-occurring entities, our approach allows the edge to indicate a specific relationship, such as 'treats,' 'contraindicates,' or 'associated with,' depending on the syntactic context.

In the following sections, we present a detailed methodology for the construction of knowledge graphs from clinical notes, describe the NLP tasks used to extract and enrich the graph's nodes and edges, and demonstrate the potential benefits of this approach through an evaluation on a dataset of medical texts. The results of this work have implications for improving the utility and interpretability of knowledge graphs in clinical applications, such as clinical decision support systems, biomedical research, and patient care.

2 Related Works

2.1 Named Entity Recognition and Relation Extraction

In knowledge graph construction, two main components are required: entities and the relationships between entities. Harnoune *et al.* in [3], built a biomedical knowledge graph using named entity recognition (NER) to identify words that serve as entities within the knowledge graph, and relation extraction (RE) to determine the relationships between those entities. The model used by Harnoune *et al.* in [3] for the NER and RE tasks is transformer-based, the most popular neural network architecture for the past few years [6]. There are BERT variants, including BioBERT, BioClinicalBERT, BioDischargeSummary, and BioRoBERTa, all of which were combined with a conditional random field

(CRF) layer for entity tag encoding. Delvin *et al.* in [7], state that the BERT model is more efficient than its predecessor in terms of learning speed and can adapt based on its initial configuration for a particular task. The best model for NER is BioClinicalBERT + CRF layer with an F1 score of 90.7%. Figure 3 illustrates the NER model architecture using the BERT variants and the CRF layer.

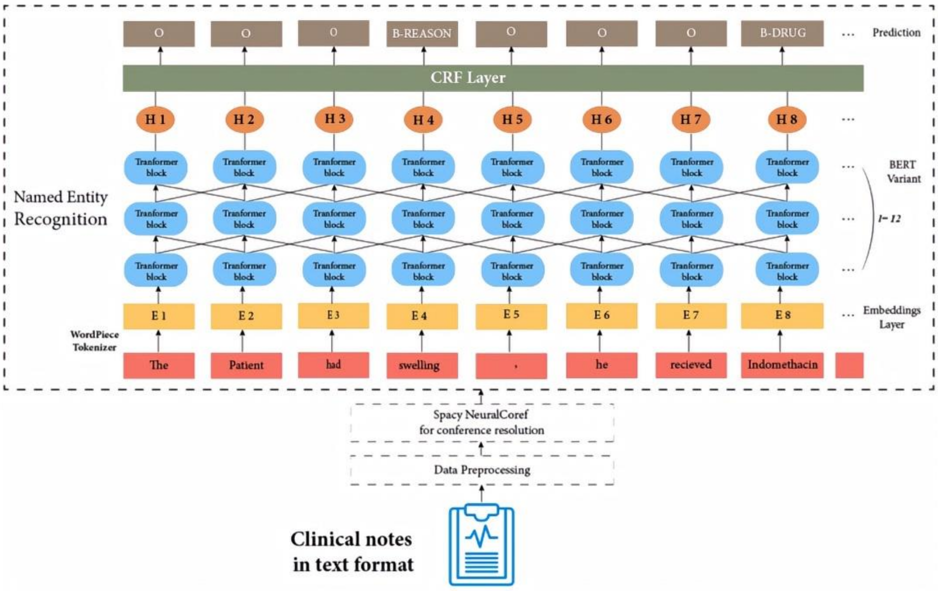

Figure 3 BERT variants and CRF layer for NER.

The BERT variant with a CRF layer is beneficial for feature extraction in the medical domain, offering advantages when dealing with lengthy reports that make feature extraction or context encoding challenging. The configuration used by Harnoune *et al.* in [3] for NER is as follows:

1. 12 stacked encoders.
2. 768 hidden size, representing the number of hidden state features in BERT.
3. 12 heads in the Multi-Head Attention layers.
4. 128 input size for the BERT neural network.
5. 17 batch size, representing the number of samples propagated through the network.
6. 10 epochs, determining how many times the learning algorithm is trained on the training dataset.
7. The CRF layer connects the obtained results with the corresponding entity classes.

Relation extraction (RE) is a binary classification problem. Harnoune *et al.* in [3], tested a BERT variant that uses the concept of sequence classification to predict relationships between entities. In sequence classification, a representation of the input sentence, called the CLS token, is obtained. This CLS token, which essentially contains information about the words and the overall context of the sentence, is fed into a fully connected neural network to perform the binary classification task. The best model for RE is BioClinicalBERT with an F1 score of 88%. The architecture of RE is shown in Figure 4.

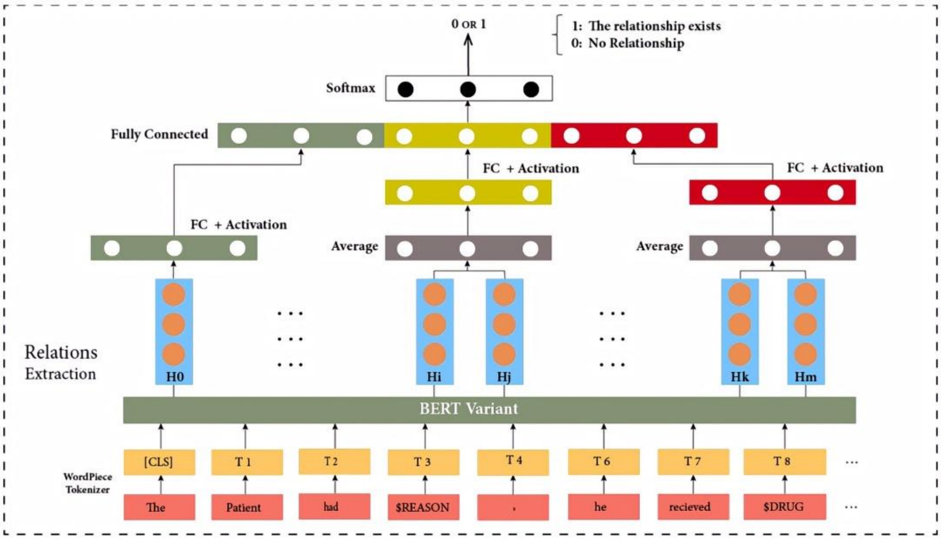

Figure 4 BERT variants for RE.

After identifying the entity names using NER and predicting the relationships between entities using RE, the entities and their relationships are structured into a triplet representation with the format (entity, relation, entity). The technique of naming relationships between entities uses the same format for two entities of the same entity type. For example, for the MEDICATION and DOSAGE entities, the relationship name is MEDICATION-DOSAGE. These triplets are then used to construct a knowledge graph.

2.2 MedicalNER

MedicalNER is a fine-tuned model from the pre-trained DeBERTaV3 for NER task on medical data [8]. MedicalNER was trained using PubMED data, a collection of medical and biological texts managed by the National Center for Biotechnology Information. The model can recognize 41 medical entities, such as SIGN_SYMPTOM, DISEASE_DISORDER, MEDICATION, DOSAGE, THERAPEUTIC_PROCEDURE, DATE, and others. DeBERTaV3 itself is an enhancement of the DeBERTa model, utilizing RTD (replaced token detection) and GDES (gradient-disentangled embedding sharing) [8,9]. In its training process, RTD involves two components: the generator and the discriminator. The generator replaces tokens that have been masked, while the discriminator determines whether a predicted token matches or is replaced from the original token. An overview of RTD is shown in Figure 5.

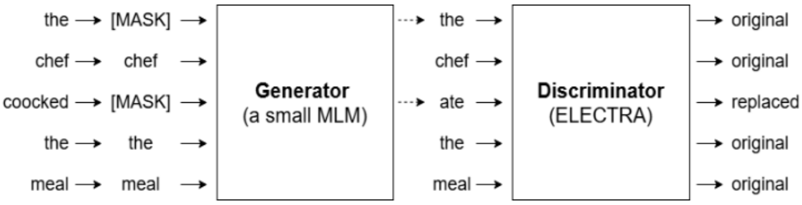

Figure 5 An overview of RTD.

During the training process, a tug-of-war can occur between the generator and the discriminator. The generator's gradients may adjust the embeddings to improve token replacement performance, but these changes can reduce the effectiveness of the embeddings for the discriminator. Conversely, the discriminator's gradients modify the embeddings to better detect replaced tokens, which can negatively affect the generator's ability to produce plausible token replacements. Therefore, Gradient-Disentangled Embedding Sharing (GDES) updates the generator's embeddings only using the Masked Language Model (MLM) loss, ensuring consistency and coherence in the generator's output. To implement GDES, He *et al.* re-parameterized the discriminator's embeddings E_D as in Eq. 1.

$$E_D =_{sg} (E_G) + E_\Delta \tag{1}$$

with stop gradient operator sg preventing the gradients from flowing through the generator embeddings E_G and only updating the residual embeddings E_Δ. An illustration of the GDES method is shown in Figure 6.

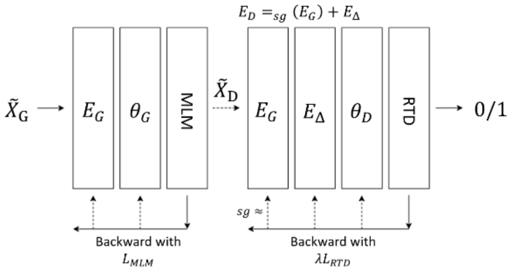

Figure 6 Illustration of Gradient-Disentangled Embedding Sharing.

2.3 Dependency Parsing and Part-of-Speech Tagging

Part-of-speech (POS) tagging is a task in natural language processing (NLP) that assigns labels to each word in a sentence with its corresponding grammatical category, such as nouns, verbs, adjectives, adverbs, and others [10]. This process is crucial and often serves as the foundation for other NLP tasks. Dependency parsing is an NLP task that analyzes grammatical structure by identifying the dependencies between words [11]. These dependencies are represented as a tree, where the nodes correspond to the words in the sentence, and the directed edges represent the dependencies between them. All words are connected to one root word, typically the main verb or predicate of the sentence.

By combining POS tagging and dependency parsing, it is possible to identify verbs that serve as the root of a sentence. Verbs that function as the root are used as the relation between entities extracted from a sentence. This technique for relation extraction was applied by Agrawal *et al.* [4] in the domain of cyber security. In their study, they constructed a knowledge graph to enhance effectiveness in cyber security education. The effectiveness of the resulting knowledge graph was evaluated using the Self-Efficacy and Metacognition Learning Inventory-Science (SEMLIS) survey developed by Thomas *et al.* in [12]. This survey was designed to assess students' perceptions of metacognition, self-efficacy, and learning processes in science. According to the survey results, 86.7% of students found the knowledge graph useful for problem-based learning tasks. Additionally, interviews were conducted to evaluate the impact of the knowledge graph qualitatively, yielding positive feedback from students.

3 Methodology

Our proposed research project flow can be seen in Figure 7.

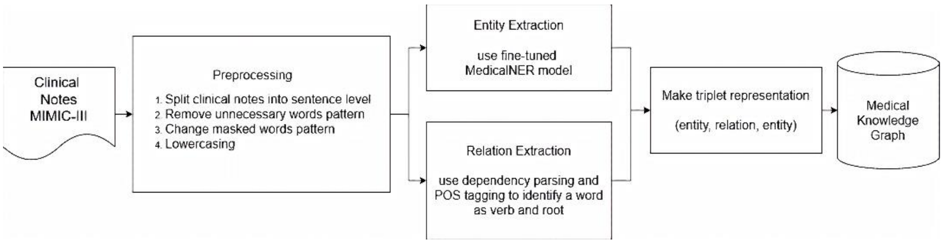
Clinical Notes MIMIC-III
Preprocessing
1. Split clinical notes into sentence level
2. Remove unnecessary words pattern
3. Change masked words pattern
4. Lowercasing
Entity Extraction
use fine-tuned MedicalNER model
Relation Extraction
use dependency parsing and POS tagging to identify a word as verb and root
Make triplet representation
(entity, relation, entity)
Medical Knowledge Graph

Figure 7 The research project work flow.

These research project phases are explained in detail in the following sections.

3.1 Dataset

This study utilized data from the Medical Information Mart for Intensive Care III (MIMIC-III) [13]. MIMIC-III is a real-world dataset comprising electronic medical records from 58,976 unique hospital admissions involving 38,597 patients in the intensive care unit (ICU) of Beth Israel Deaconess Medical Center between 2001 and 2012. The dataset includes free-text data that serves as medical records for each patient. The MIMIC-III dataset has been employed in several studies. Zhu *et al.* [14] used MIMIC-III data to develop a predictive model for patient mortality rates among those requiring mechanical ventilation in the ICU. Aldughayfiq *et al.* [15] captured semantic relationships in MIMIC-III using knowledge graphs. Gong *et al.* [2] utilized the dataset to build a safe medication recommendation system.

3.2 Preprocessing

The training data used for fine-tuning the MedicalNER model consists of a combination of the Medical Entity Recognition data from Kaggle and a subset of medical records from MIMIC-III. The Medical Entity Recognition data is in the form of text files, where each line contains a single word along with its label, separated by a tab. Each sentence is separated by a blank line. The data already includes labels such as O, T, and D, but there is no further explanation regarding the meaning of these labels. The medical records from MIMIC-III, on the other hand, are unstructured text stored in a column of a CSV file, with each row representing the medical record of a patient identified by a unique ID. These records are free-text data stored in a single column. The MIMIC-III data was preprocessed by splitting it into sentences, lowercasing, and either removing or modifying identified word patterns. An example of this preprocessing is shown in Table 1.

Table 1 Preprocessing training data for fine-tuning.

Preprocess	Text	Result
Lowercasing	PAST MEDICAL HISTORY	Past medical history
Removing unnecessary word patterns	[**Hospital1 69**]	-
Modifying masked word patterns	[**2112-6-5**]	2112-6-5

Subsequently, the MIMIC-III data used for training was reformatted to match the format of the Medical Entity Recognition data. The labeling process was done manually by the researchers, who analyzed the meaning and semantics of each word with the help of large language model (LLMs). The labels used were limited to SIGN_SYMPTOM, DISEASE_DISORDER, MEDICATION, DOSAGE, THERAPEUTIC_PROCEDURE, and DATE. Entities outside these categories were labeled as O (Others).

3.3 Entity and Relation Extraction

Each preprocessed sentence is analyzed to extract the entities and relations it contains. MedicalNER is used to label words that represent medical entities of interest. Dependency parsing is applied to assign labels to each word based on its grammatical role, such as verb, noun, adjective, and others. POS tagging is used to label each word according to its grammatical function, such as root, subject, object, and others. The labels obtained from dependency parsing and POS tagging are further filtered to identify a word that functions as the verb and root of the sentence. This word is used to define the relation between entities.

For example, from the sentence "she was continued on diltiazem for rate control," MedicalNER will label 'diltiazem' as MEDICATION and the other tokens as Others. Let's say 'she' in this sentence refers to 'patient_123,' then this will be considered as the PATIENT entity. Furthermore, using dependency parsing and POS tagging, the word 'continued' is identified as the root and verb of the sentence. Therefore, 'continued' will be used as the relation between patient_123 and diltiazem. The name of the relation can also be supplemented with a description of the types of entities being connected, for instance, in this case, it becomes 'continued medication.'

3.4 Knowledge Graph Construction

After identifying the entities and relationships contained in all sentences, these entities and relationships are used to create triplets representation. A triplet representation is formed in the following format (entity, relation, entity). The triplet produced from the example in Section 3.3 is as follows: (patient_123, continued medication, diltiazem). Those triplets are used to construct a medical knowledge graph. The constructed knowledge graph is evaluated both quantitatively and qualitatively at the triplet level. For example, triplets in Table 2 are constructed to the knowledge graph in Figure 8.

Table 2 Example of entity–relation–entity triplets used to construct the knowledge graph from clinical note data.

Triplets
(Patient_123, diagnosed, stroke)
(Patient_123, started medication, diltiazem)
(aspirin, dose for patient_123, 30 mg)
(Patient_123, had procedure, rigid bronchoscopy)

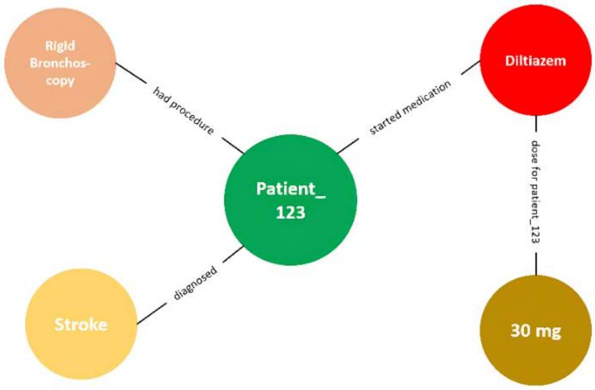

Figure 8 Example of knowledge graph construction from triplets.

4 Results and Evaluation

The performance of Medical NER model was tested using medical record data from MIMIC-III. The test data was labeled manually with 100 randomly selected sentences from the MIMIC-III dataset. Based on the evaluation, the Medical NER model was found to be insufficient in accurately identifying entities from MIMIC-III medical record data. Therefore, additional fine-tuning was conducted to improve the model's performance on the test data.

The fine-tuning process used the Medical Entity Recognition dataset from Kaggle. This dataset contains a variety of sentences related to the medical domain, but the provided labels are limited to single-letter annotations without further explanation. Therefore, 1,076 sentences or 19,600 tokens were manually labeled by the researcher. These sentences were split into 80% training data and 20% validation data. The best parameters for fine-tuning were found to be a learning rate of 2e-5, 5 epochs, and a batch size of 32.

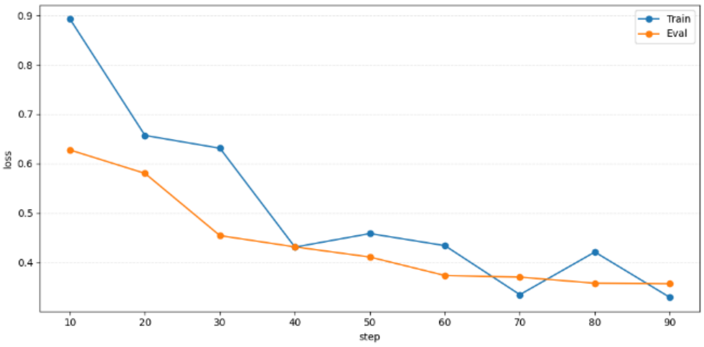

Figure 9 Training and validation loss.

The learning curve shows a rapid decrease in both training and validation loss during the early phase of fine-tuning (steps 10-30), indicating that the model quickly learns useful representations from the training data. After this initial phase, both curves continue to decline more gradually and approach a plateau (validation loss around 0.36). The validation loss remained close to, and slightly above, the training loss throughout most of the run, which indicates acceptable generalization and no strong evidence of runaway over fitting within the logged steps. A comparison of evaluations before and after fine-tuning can be seen in Table 3 and Table 4.

Table 3 MedicalNER evaluation on test data.

Entity	Precision	Recall	F1-score
Overall	**0.55**	**0.67**	**0.60**
DATE	0.26	0.45	0.33
DISEASE_DISORDER	0.63	0.72	0.67
DOSAGE	0.19	0.40	0.26
MEDICATION	0.74	0.81	0.77
SIGN_SYMPTOM	0.60	0.62	0.61
THERAPEUTIC_PROCEDURE	0.66	0.73	0.69

Table 4 Fine-tuned MedicalNER evaluation on test data.

Entity	Precision	Recall	F1-score
Overall	**0.83**	**0.86**	**0.84**
DATE	0.77	0.91	0.83
DISEASE_DISORDER	0.79	0.90	0.84
DOSAGE	0.52	0.54	0.53
MEDICATION	0.90	0.93	0.92
SIGN_SYMPTOM	0.92	0.90	0.91
THERAPEUTIC_PROCEDURE	0.78	0.79	0.78

After fine-tuning, the model's performance improved significantly. The overall F1-score for all entities increased by 0.24, from 0.6 to 0.84. The extracted entities were subsequently used for knowledge graph construction.

The data used for the experiment were samples of medical records from 20 randomly selected patients. These records yielded 1,711 sentences and produced 854 triplets. Those triplets were utilized for constructing a knowledge graph. An example of the knowledge graph result for one patient can be seen in Figure 9.

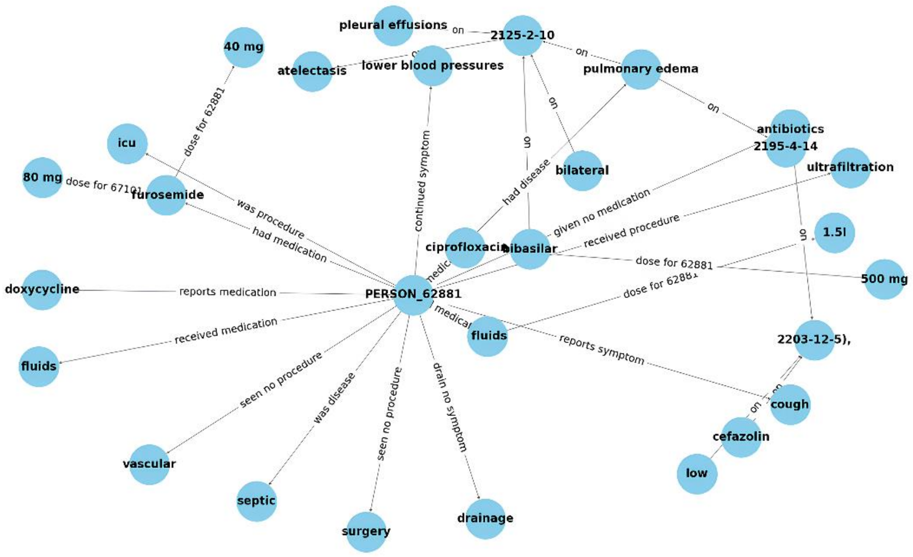

Figure 10 Medical knowledge graph for one patient.

The total of 1,711 sentences was also converted into triplets manually as gold labels for evaluating the predicted triplets. The constructed triplets were evaluated both quantitatively and qualitatively [15]. The quantitative evaluation showed good results, reaching an F1-score of 0.91. Subsequently, a qualitative evaluation was conducted by an expert with a background in both medicine and informatics. The expert did not have access to the source data, thereby enabling a more objective qualitative evaluation. This evaluation assessed the triplets based on two aspects: correctness and informativeness, which were compared to triplets constructed by Harnoune *et al.* in [3], with the scores were between 1 to 5 for each triplet.

Table 5 Quantitative evaluation.

Precision	**Recall**	**F1-score**
0.89	0.93	0.91

Table 6 Qualitative evaluation.

Correctness	**Informativeness**
4.2 / 5	3.8 / 5

Based on the results of the quantitative evaluation, the generated medical knowledge graph performed well, achieving an F1-score of 0.91. In the quantitative evaluation, there were two types of errors, false positives and false negatives. A false positive occurs when the predicted triplet is incorrect or not present in the test data. This can result from errors in entity prediction. For instance, from the sentence "two days prior to admission, she was started on a prednisone taper and one day prior to admission she required oxygen at home in order to maintain oxygen saturation greater than 90%," the NER model incorrectly predicted 'oxygen' as a THERAPEUTIC_PROCEDURE, whereas it should be classified as Others since it is outside the scope of extracted entities. On the other hand, a false negative occurs when the model fails to generate a triplet that is present in the test data. This error can also be attributed to the NER model failing to identify entities or due to the sentence structure being informal, making the subject-predicate-object structure ambiguous. For example, from sentence "she had fullness in ear and she also had a cold coinciding to the onset of her headache," the NER model did not predict 'fullness in ear' as a SIGN_SYMPTOM, leading to the triplet not being formed.

Meanwhile, in the qualitative evaluation conducted by an expert, a score of 4.2 out of 5 was obtained for correctness, and 3.8 out of 5 for informativeness. For correctness evaluation, the majority of errors occurred in entity extraction, where phrases were only partially extracted, capturing only one word of the phrase. For example, from the sentence "he remained asymptomatic with no shortness of breath or chest pain after transfer," the NER model only predicted "pain" as SIGN_SYMPTOM, which should have been "chest pain." This issue may be attributed to a lack of training data containing entity names in the form of phrases. Regarding informativeness, in some cases, relationships expressed using natural word were more effective than template-based relations but also introduced ambiguity. For instance, a relationship between the entities DISEASE_DISORDER and DATE labeled as 'on' could lead to ambiguity. It is unclear whether the disease first occurred on that date or if the patient had already recovered by that date.

5 Conclusion and Future Work

Based on the experiments and analyses conducted, fine-tuned MedicalNER is capable of predicting entities from medical record data more effectively compared to before fine-tuning, achieving an F1-score of 0.84. The triplets that extracted using NER, POS tagging, and dependency parsing for medical knowledge graph construction demonstrated strong performance in the quantitative evaluation with an F1-score of 0.91. In the qualitative evaluation, a score of 4.2 out of 5 was obtained for correctness, and 3.8 out of 5 for informativeness. This system can be utilized to build a medical knowledge graph with meaningful and contextual relationships, aiding doctors in analyzing each patient and supporting decision-making processes.

For future work, MedicalNER model can be developed with a dataset tailored to specific needs. Although both use medical datasets, different datasets may vary in writing style and the entities discussed within them, which can result in the model making incorrect predictions. Another idea is developing a machine learning or deep learning model that can directly predict the relationships between entities based on the semantics of the sentence.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Mrs. Masayu, for her invaluable guidance and support throughout this project. I also thank my family and girlfriend for their constant encouragement and patience.

References

[1] Nicholson, D.N. & Greene, C.S., *Constructing Knowledge Graphs and Their Biomedical Applications,* Computational and Structural Biotechnology Journal, **18**, 1414-1428, 2020.

[2] Gong, F., Wang, M., Wang, H., Wang, S. & Liu, M., *SMR: Medical Knowledge Graph Embedding for Safe Medicine Recommendation,* Big Data Research, **23**, 100174, 2021.

[3] Harnoune, A., Rhanoui, M., Mikram, M., Yousfi, S., Elkaimbillah, Z. & Asri, B.E., *BERT Based Clinical Knowledge Extraction for Biomedical Knowledge Graph Construction and Analysis,* Computer Methods and Programs in Biomedicine Update, **1**, 100042, 2021.

[4] Agrawal, G., Deng, Y., Park, J., Liu, H. & Chen, Y.C., *Building Knowledge Graphs from Unstructured Texts,* Applications and impact analyses in cyber security education. Information, **13**(11), p.526, 2022.

[5] Shi, L., Li, S., Yang, X., Qi, J., Pan, G. & Zhou, B., *Semantic Health Knowledge Graph: Semantic Integration of Heterogeneous Medical Knowledge and Services,* BioMed research international, 2017.

[6] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. & Davison, J., *Transformers: State-of-the-art Natural Language Processing,* In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, 38-45, 2020.

[7] Kenton, J.D.M.W.C. & Toutanova, L.K., *Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding,* In Proceedings of naacL-HLT, **1**, p. 2, June, 2019.

[8] He, P., Gao, J. & Chen, W., *Debertav3: Improving Deberta using Electra-style Pre-training with Gradient-disentangled Embedding Sharing.* arXiv preprint arXiv, 2111,09543, 2021.

[9] Clark, K., Luong, M., Le, Q. & Manning, C., *Electra: Pre-training Text Encoders as Discriminators Rather than Generators,* arXiv preprint arXiv, 2003, 10555, 2020.

[10] Chiche, A. & Yitagesu, B., *Part of Speech Tagging: A Systematic Review of Deep Learning and Machine Learning Approaches,* Journal of Big Data, **9**(1), p.10, 2022.

[11] Nivre, J., De Marneffe, M.C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N. & Tsarfaty, R., *May. Universal Dependencies V1: A Multilingual Treebank Collection,* In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pp. 1659-1666, 2016.

[12] Thomas, G., Anderson, D. & Nashon, S., *Development of an Instrument Designed to Investigate Elements of Science Students' Metacognition, Self-efficacy and Learning Processes,* The SEMLI-S. Int. J. Sci. Educ., **30**, 1701–1724, 2008.

[13] Johnson, A.E., Pollard, T.J., Shen, L., Lehman, L.W.H., Feng, M., Ghassemi, M., & Mark, R.G., *MIMIC-III, a Freely Accessible Critical Care Database,* Scientific data, **3**(1), 1-9, 2016.

[14] Zhu, Y., Zhang, J., Wang, G., Yao, R., Ren, C., Chen, G. & Yu, Q., *Machine Learning Prediction Models for Mechanically Ventilated Patients: Analyses of the Mimic-iii Database,* Frontiers In Medicine, **8**, 662340, 2021.

[15] Aldughayfiq, B., Ashfaq, F., Jhanjhi, N.Z. & Humayun, M., Capturing semantic relationships in electronic health records using knowledge graphs: An implementation using mimic iii dataset and graphdb, Healthcare 11,(12), p. 1762, MDPI, 2023.

[16] Gao, J., Li, X., Xu, Y.E., Sisman, B., Dong, X.L. & Yang, J., *Efficient Knowledge Graph Accuracy Evaluation,* arXiv preprint arXiv, 1907.09657, 2019.