Innovations in Domain Knowledge Augmentation of Contextual Models

by

Georgios Michalopoulos

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2022

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:
Steven Bethard
Associate Professor,
School of Information
University of Arizona

Supervisor(s):
Ian McKillop
Associate Professor,
David R. Cheriton School of Computer Science
University of Waterloo
Helen Chen
Professor of Practice, School of Public Health Sciences
Cross-Appointed from
the David R. Cheriton School of Computer Science
University of Waterloo

Internal Member:
Jimmy Lin
Professor,
David R. Cheriton School of Computer Science
University of Waterloo

Internal Member:
Yaoliang Yu
Assistant Professor,
David R. Cheriton School of Computer Science
University of Waterloo

Internal-External Member: **Alexander Wong**
Professor,
Systems Design Engineering
University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

I am the sole author for Chapters 1 and 6 which are written under the supervision of Drs. Chen and McKillop. These two chapters have not been published.

This thesis consists in part of four manuscripts that have been published, or under review with co-authors. I am the lead author in each publication, and the contribution is listed as follows:

Research presented in Chapter 2:

I conducted this research at the University of Waterloo under the supervision of Drs. Chen and McKillop. I designed the study with advice from Drs. Chen and Wong. I created and completed the experiments. I drafted the manuscript, and all co-authors contributed to the revision of the manuscript.

George Michalopoulos, Ian McKillop, Alexander Wong, and Helen Chen. 2022. LexSubCon: Integrating Knowledge from Lexical Resources into Contextual Embeddings for Lexical Substitution. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1226–1236, Dublin, Ireland. Association for Computational Linguistics. DOI:10.18653/v1/2022.acl-long.87

Research presented in Chapter 3:

I conducted this research at the University of Waterloo under the supervision of Drs. Chen and McKillop. I designed the study with advice from Drs. Chen and Wong. I created and completed the experiments with assistance from Yuanxin Wang and Hussam Kaka. I drafted the manuscript, and all co-authors contributed to the revision of the manuscript.

George Michalopoulos, Yuanxin Wang, Hussam Kaka, Helen Chen, and Alexander Wong. 2021. UmlsBERT: Clinical Domain Knowledge Augmentation of Contextual Embeddings Using the Unified Medical Language System Metathesaurus. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1744–1753, Online. Association for Computational Linguistics. DOI:10.18653/v1/2021.naacl-main.139

Research presented in Chapter 4:

I conducted this research during my internship at Microsoft under the supervision of Dr. Williams. I designed the study with consultations from Drs. Williams and Lin, and Gagandeep Singh. I drafted the manuscript and each co-author provided intellectual input on the manuscript.

George Michalopoulos, Kyle Williams, Gagandeep Singh and Thomas Lin. 2022.MedicalSum: A Guided Clinical Abstractive Summarization Model for Generating Medical Reports from Patient-Doctor Conversations.

Research presented in Chapter 5:

I conducted this research at the University of Waterloo under the supervision of Drs. Chen and McKillop. I designed the study with input from Drs. Chen and Wong, as well as from Michal Malyska and Nicola Sahar at Sementic Health Inc. I designed and implemented the experiments and drafted the manuscript. All co-authors contributed to the revision of the manuscript.

George Michalopoulos, Michal Malyska, Nicola Sahar, Alexander Wong, and Helen Chen. 2022. ICDBigBird: A Contextual Embedding Model for ICD Code Classification. In Proceedings of the 21st Workshop on Biomedical Language Processing, pages 330–336, Dublin, Ireland. Association for Computational Linguistics. DOI:10.18653/v1/2022.bionlp-1.32

Abstract

The digital transformation of our society is creating a tremendous amount of data at an unprecedented rate. A large part of this data is in unstructured text format. While enjoying the benefit of instantaneous data access, we are also burdened by information overload. In healthcare, clinicians have to spend a significant portion of their time reading, writing and synthesizing data in electronic patient record systems. Information overload is reported as one of the main factors contributing to physician burnout; however, information overload is not unique to healthcare. We need better practical tools to help us access the right information at the right time. This has led to a heightened interest in high-performing Natural Language Processing research and solutions.

Natural Language Processing (NLP), or Computational Linguistics, is a sub-field of computer science that focuses on analyzing and representing human language. The most recent advancements in NLP are large pre-trained contextual language models (e.g., transformer based models), which are pre-trained on massive corpora, and their context-sensitive embeddings (i.e., learned representation of words) are used in downstream tasks. The introduction of these models has led to significant performance gains in various downstream tasks, including sentiment analysis, entity recognition, and question answering. Such models have the ability to change the embedding of a word based on its imputed meaning, which is derived from the surrounding context.

Contextual models can only encode the knowledge available in raw text corpora. Injecting structured domain-specific knowledge into these contextual models could further improve their performance and efficiency. However, this is not a trivial task. It requires a deep understanding of the model's architecture and the nature and structure of the domain knowledge incorporated into the model. Another challenge facing NLP is the "low-resource" problem, arising from a shortage of publicly available (domain-specific) large datasets for training purposes. The low-resource challenge is especially acute in the biomedical domain, where strict regulation for privacy protection prohibits many datasets from being publicly available to the NLP community. The severe shortage of clinical experts further exacerbates the lack of labeled training datasets for clinical NLP research.

We approach these challenges from the knowledge augmentation angle. This thesis explores how knowledge found in structured knowledge bases, either in general-purpose lexical databases (e.g., WordNet) or domain-specific knowledge bases (e.g., the Unified Medical Language Systems or the International Classification of Diseases), can be used to address the low-resource problem. We show that by incorporating domain-specific prior knowledge into a deep learning NLP architecture, we can force an NLP model to learn the associations

between distinctive terminologies that it otherwise may not have the opportunity to learn due to the scarcity of domain-specific datasets.

Four distinct yet complementary strategies have been pursued. First, we investigate how contextual models can use structured knowledge contained in the lexical database WordNet to distinguish between semantically similar words. We update the input policy of a contextual model by introducing a new mix-up embedding strategy for the input embedding of the target word. We also introduce additional information, such as the degree of similarity between the definitions of the target and the candidate words. We demonstrate that this supplemental information has enabled the model to select candidate words that are semantically similar to the target word rather than those that are only appropriate for the sentence's context.

Having successfully proven that lexical knowledge can aid a contextual model in distinguishing between semantically similar words, we extend this approach to highly specialized vocabularies such as those found in medical text. We explore whether using domain-specific (medical) knowledge from a clinical Metathesaurus (UMLS Metathesaurus) in the architecture of a transformer-based encoder model can aid the model in building 'semantically enriched' contextual representations that will benefit from both the contextual learning and the domain knowledge. We also investigate whether incorporating structured medical knowledge into the pre-training phase of a transformer-based model can incentivize the model to learn more accurately the association between distinctive terminologies. This strategy is proven to be effective through a series of benchmark comparisons with other related models.

After demonstrating the effect of structured domain (medical) knowledge on the performance of a transformer-based encoder model, we extend the medical features and illustrate that structured medical knowledge can also boost the performance of a (medical) summarization transformer-based sequence-to-sequence model. We introduce a guidance signal consisting of the medical terminologies in the input sequence. Moreover, the input policy is modified by utilizing the semantic types from UMLS, and we also propose a novel weighted loss function. Our study demonstrates the benefit of these strategies in providing a stronger incentive for the model to include relevant medical facts in the summarized output.

We further examine whether an NLP model can take advantage of both the relational information between different labels and contextual embedding information by introducing a novel attention mechanism (instead of augmenting the architecture of contextual models with structured information as described in the previous paragraphs). We tackle the challenge of automatic ICD coding, which is the task of assigning codes of the International

Classification of Diseases (ICD) system to medical notes. Through a novel attention mechanism, we integrate the information from a Graph Convolutional Network (GCN) that considers the relationship between various codes with the contextual sentence embeddings of the medical notes. Our experiments reveal that this enhancement effectively boosts the model's performance in the automatic ICD coding task.

The main contribution of this thesis is two-fold: (1) this thesis contributes to the computer science literature by demonstrating how domain-specific knowledge can be effectively incorporated into contextual models to improve model performance in NLP tasks that lack helpful training resources; and (2) the knowledge augmentation strategies and the contextual models developed in this research are shown to improve NLP performance in the biomedical field, where publicly available training datasets are scarce but domain-specific knowledge bases and data standards have achieved a wide adoption in electronic medical records systems.

Acknowledgements

I would like to express my gratitude and appreciation to my advisors: Prof. Helen Chen and Prof. Ian McKillop.

Helen was my advisor and my mentor. Thanks to her guidance and support, I was able to find the research projects that I became passionate about; I learned how to ask the right research questions and how to properly present my work; I learned to work as a member of a team and to take personal responsibility for my work. She helped me to become a better researcher, and I am sure her work ethics will continue to guide me in my later career.

I cannot thank enough Ian for his patience and for the invaluable advice he gave me all along my Ph.D. project, helping me to never lose sight of the end goal. Whenever I was in doubt, he generously shared his academic experience with me.

I also warmly thank my advisory committee members Prof. Alexander Wong, Prof. Jimmy Lin and Prof. Yaoliang Yu for their support and advice throughout my PhD. I would also like to offer my special thanks to Prof. Steven Bethard for serving as the External Examiner and for spending time reading and providing valuable comments on my thesis.

On a more personal note, I am extremely grateful to my family. My parents, Eleni and Michalis, were always there for me with their unconditional love and support. Their life choices inspire me to become a better person. I also want to thank my grandparents who always had an encouraging word to say.

Last, but not least, I want to thank my amazing wife, Marianna, for her patience and support, for her company and the countless cups of coffee she made on the nights when I was in front of my computer frustrated about a buggy code. When sometimes I lost faith in myself or was not sure about the path I should follow, she was there for me to encourage and help me sort things out.

Dedication

This is dedicated to the loving memory of my mother, Eleni. She was always there for me, and her support made me the person that I am today. Even if she was taken from us too soon, I hope that she will be always watching over me and be proud of her son.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Living in an increasingly digital society, we generate and consume a large amount of data daily. Even though we have already accumulated a tremendous volume of data in private and public information systems, our desire and need for more data will continue to grow [2]. The COVID-19 pandemic has further accelerated digitalization in nearly every industry, and we are now generating data at an unprecedented rate. Having access to this much data is a double-edged sword. On the one hand, accessing and sharing information enables us to make informed decisions and work collaboratively; on the other hand, navigating such a large amount of data to locate precise information is extremely time-consuming. In healthcare, a clinician needs to spend a significant portion of their time reading, writing and synthesizing data in electronic patient record systems [107]. The documentation requirements for electronic health records (EHR) have been shown to be a significant factor contributing to physician burnout [112]. We need practical tools that can handle data for us - to search, organize, visualize, translate, summarize and, ultimately, have the correct information at the right time, instantaneously.

A significant part of the data currently available is in text format, such as news, books and scientific publications, legal and medical reports and social media posts, to name a few. Automating the processing and understanding of text data requires Natural Language Processing (NLP), or Computational Linguistics techniques that focus on the analysis and representation of human language [121]. Some of the major research areas that are part of NLP include (i) machine translation (i.e., the translation of text from one language to another without human interference [42]); (ii) automatic summarization (i.e., creating a summary that contains the most important information derived from input text [26]) and (iii) text classification (i.e., the assignment of a set of predefined classes to a set of documents [75]).

NLP research has made significant advancements in recent years. Some of the most recent trends in NLP include [104]: (i) Transfer learning: leveraging data from additional domains or tasks to train highly accurate models [92]; (ii) Knowledge augmentation: incorporating external knowledge to provide comprehensive relational information [63] in order to enhance the reasoning of pre-trained language models [120] and (iii) Low-resource NLP tasks: constructing accurate models for NLP tasks that lack useful training resources such as labeled data or number of experts [64].

This dissertation focuses on knowledge augmentation: i.e., augmenting NLP models with knowledge obtained from structured knowledge bases like WordNet [73] and biomedical knowledge bases like UMLS [18] to tackle the low-resource NLP challenge. We explore the strategies of incorporating domain-specific prior knowledge into a deep learning architecture to force an NLP model to learn the associations between distinctive terminologies, which it otherwise may not have the opportunity to learn, due to the scarcity of domain specific datasets, particularly in the biomedical domain. We also examine the effectiveness of these strategies in improving the performance and generalization of contextual models.

1.1 Evolution of Natural Language Processing Models

The birth of NLP can be traced back to the 1940s. NLP originally focused on machine translation, where words were mapped from one language to another using predefined dictionaries [48]. In the 1990s, machine learning models were used to infer probabilities instead of hard-coded syntactic rules from massive datasets. These models transformed the input text to numerical data by using common feature engineering methods like Bag of Words (BoW), where a set of vectors containing the count of word occurrences in the document were created, or Term Frequency-Inverse Document Frequency (TF-IDF), where each word count was divided by the number of documents that each word appears in. These transformations enabled tabular data models, such as support vector machines [38], and regression to be employed for different NLP tasks, for example, the text categorization task [12]. However, the models mentioned above were unable to assess the dependencies between the words.

Recurrent Neural Networks (RNN) [17] could learn the dependencies between words by taking into consideration not only the input text, but also the output of the previous layer. RNN models created hidden states (memory) at each step to maintain the information calculated in the previous steps, using previous outputs and the current token as inputs.

The main disadvantage of RNN models was that during back-propagation, the gradient at each output depended on the calculation of the current and the previous steps, thus introducing the exploding/vanishing gradient problems for long-term dependencies. By using a gating mechanism, a popular variant of RNN, Long Short Term Memory Networks (LSTM)[40] circumvented the vanishing gradient problem. The main difference between an LSTM and a vanilla RNN was that the cell state of the LSTM was regulated by a structure called a gate, with each gate consisting of a pointwise multiplication operation and a sigmoid layer (Figure 1.1). By learning the parameters of its gate, the model could then acquire a better understanding of the input sentence. Recurrent models were widely applied in different sequence modeling problems, such as machine translation and language modeling [10, 101]. RNN models used word embedding as their main structured representation of words. Word embeddings were numerical vectors with fixed dimensionality and whose relative geometrical positions reflect similarity properties of the embedded words [78]. However, traditional word embedding methods such as Word2vec [72] produced a constant context-independent vector representation for each word. Therefore, they could not distinguish between a word's different meanings in a given context. The contextualized word embeddings in a bidirectional language model (ELMo) introduced by Peters et al. [82] extended traditional word embeddings to learn context-sensitive features by changing the embedding of a word based on its imputed meaning thus achieving the state-of-the-art for major NLP benchmarks including sentiment analysis [97] and question answering [85].

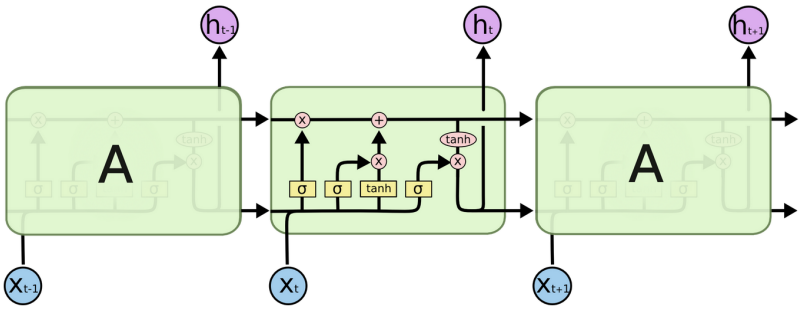


Figure 1.1: LSTM architecture in [109].

Sequence-to-sequence models are a special class of RNN architectures that map the input sequence to the output sequence [101]. These models were composed of an encoder and a decoder. The task of an encoder network was to understand the input sequence and then generate a compact representation. With such representation at hand, the decoder could generate a target sequence. Bahdanau et al. [10] introduced the 'attention' mechanism to enable the decoder to focus on the relevant parts of an input sequence. The main difference

between an attention model and a vanilla sequence-to-sequence model was that, with the former, the encoder passed all the hidden states to the decoder instead of only passing the last hidden state. As a result, the main advantage of the attention mechanism was that the decoder could take into consideration all the hidden states to generate a context vector for each time step.

However, the sequential nature of these models precluded parallelization in training process, which is a critical obstacle in processing long sequences.

1.2 Self-Attention Models

Self-attention is an extension of the attention mechanism relating different positions of a single sequence in order to compute a representation of the same sequence [113].

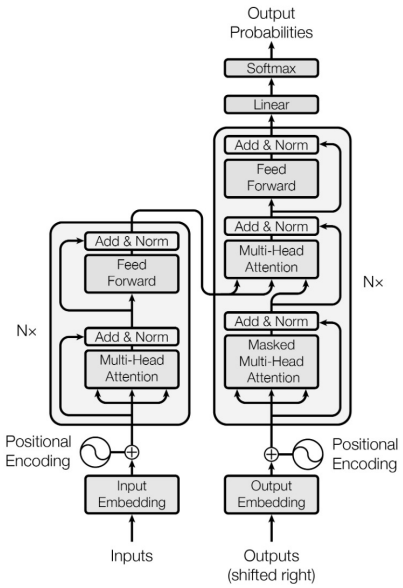


Figure 1.2: The Transformer architecture in [113].

Transformer models [113] used self-attention layers to take into consideration the other positions/words in the input sequence while encoding a particular word and have become a key component in recent language models. The main advantages of the self-attention mechanism were that the training speed of a model could be boosted, and a better word

representation could be constructed. As a result, state-of-the-art performance could be achieved in NLP tasks (such as machine translation [113]). In the original Transformer paper [113] (Figure 1.2), the encoding component of a Transformer model was composed of a stack of six encoders, with each encoder containing two sub-layers: a self-attention layer and a feed-forward layer (followed by a normalization layer). The decoding component comprised a self-attention layer, a cross-attention block with the encoded input, and a feed-forward layer. The key characteristic of a Transformer model was its self-attention mechanism. For each input vector, a Value vector V, a Query vector Q, and a Key vector K of dimension d_k were created by multiplying three matrices with the corresponding embedding. The self-attention score was then calculated using these vectors to rank each word against the whole sequence (equation 1.1).

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1.1}$$

The ranking determined which part of the sequence should get more attention when encoding certain words.

Language models like Bidirectional Encoder Representations from Transformers (BERT) [23] took advantage of the Transformer's architecture by using its encoder component. The key idea behind BERT was that, by utilizing the bidirectional training of Transformers for language modeling, the model could gain a deep sense of the context as the model could learn the context of a word based on its surroundings. Thus, BERT achieved the state-of-the-art for major NLP tasks, including language inference [117] and text classification [115]. The pre-training of the BERT model was done on massive corpora, and the context-sensitive embeddings could be further fine-tuned for a downstream task by integrating them into a task-specific architecture. In Figure 1.3, we provide examples of pre-training and downstream tasks of an encoder and an encoder-decoder architecture. One of the main limitations of transformer models like BERT [23] was the quadratic dependency on the sequence length due to their full attention mechanism. By introducing a sparse attention mechanism, models like BigBird [122] and LongFormer [15] overcame these limitations and allowed the processing of lengthier documents.

These contextual models have significantly improved some major NLP benchmarks, including sentiment analysis [97] and question answering [85]. However, they can only encode the knowledge available in raw text corpora, thus still retaining some of the limitations of traditional static embeddings [57]. Incorporating external knowledge, particularly, structured domain-specif knowledge into these contextual models could further improve their performance and efficiency.

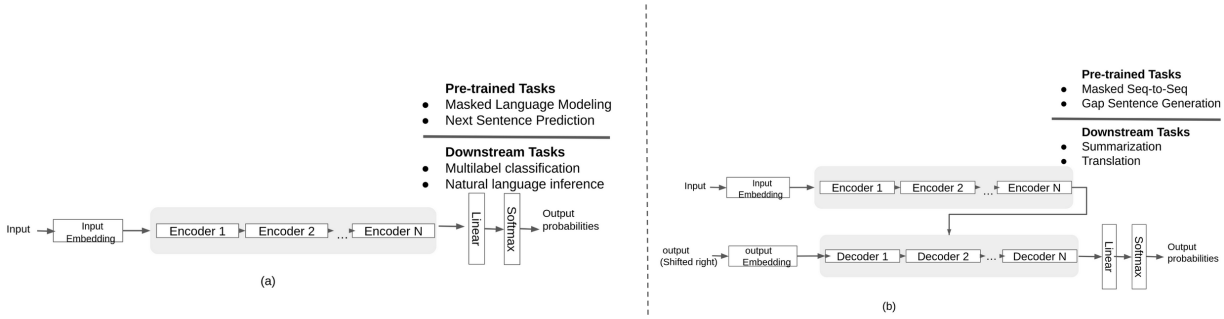


Figure 1.3: Examples of pre-training and downstream tasks of **(a)** an encoder and **(b)** an encoder-decoder architecture.

1.2.1 Augmentation of Contextual Models with General Lexical Knowledge

There have been multiple attempts to augment contextual models with external lexical knowledge. WordNet is an extensive English lexical database with words that are grouped into sets of cognitive synonyms (synsets), where each synset expresses a distinct concept [73]. WordNet uses conceptual-semantic and lexical relations to create meaningful connections between the different synsets.

One notable approach of augmenting contextual model with WordNet is Sense-BERT [59]. Sense-BERT was pre-trained to predict each word's supersenses (i.e., semantic classes). The prediction of the supersenses was achieved by incorporating lexical semantics from the lexical database WordNet into the model's pre-training objective and adding supersense information to the input embedding. A subsequent attempt called GlossBERT [41] focused on improving word sense disambiguation by using context-gloss pairs on the standard sentence-pair classification task of a BERT model. Further enhancements were made in LiBERT [57] by using the synonyms and direct hyponym-hypernym pairs knowledge to improve its performance. It introduced an additional task into the pre-training phase of the model to recognize a semantic relation between a word pair. The LiBERT model demonstrated that complementing contextual information with lexical knowledge were beneficial for multiple NLP tasks (e.g., sentence classification and sentence-pair regression) [57]. In this thesis, WordNet is employed to examine the benefits of incorporating external knowledge in lexical substitution tasks.

Lexical Substitution [66] is the task of generating appropriate words which can replace a target word in a given sentence without changing the sentence's meaning. The increased research interest in Lexical Substitution is due to its utility in various NLP fields. These

include data augmentation, which is the task of the artificial creation of training data [13] and paraphrase generation, which is the task of creating new texts that convey the same meaning as the original sentence while using different words or sentence structures [126].

Garí Soler et al. [31] used ELMo in the lexical substitution task by calculating the cosine similarity between the ELMo embedding of the target word and all the candidate substitutes. Other scientists, such as Zhou et al. [127], enhanced contextual models in the lexical substitution task by improving the BERT's standard procedure of the masked language modeling task.

However, these models did not consider incorporating structured knowledge from external lexical databases into their prediction process. These lexical resources can boost the model's performance by providing additional information, such as the definitions of the target and candidate words. Having access to the definition helps ensure that the candidate word is semantically similar to the target word, rather than just being appropriate for the sentence's context. External lexical resources can also aid by enriching the proposed candidate word list beyond the vocabulary of the contextual model.

1.2.2 Augmentation of Contextual Models with Biomedical Domain Knowledge

In addition to general lexical knowledge, domain-specific knowledge could further help contextual models to understand lexical and semantic relations used in a specific field, such as in biomedical languages.

There are several challenges facing biomedical NLP tasks, including the complexity of biomedical language, the frequency of typing or spelling errors and the heterogeneous formats of clinical documents across biomedical subdomains and health institutions [44]. Also, many of the publicly available data sets are specific to certain clinical disciplines or clinical settings, which results in limited generalizability of clinical NLP models [100].

Biomedical Knowledge Bases

Fortunately, structured knowledge resources, such as terminology, ontology, and medical codes, are abundant and well established in the biomedical domain. There are multiple international standards of structured knowledge bases of medical information. The International Statistical Classification of Diseases and Related Health Problems (ICD) is a widely-adopted clinical vocabulary system used in healthcare settings to collect patient care and hospital operation data. Initially, ICD was used solely to collect mortality data [76]. In response to a growing demand for more detailed clinical data, the original ICD

taxonomy has been updated in subsequent versions with the ability to capture more detailed data than its predecessor. Also, the National Library of Medicine has introduced and maintains the Medical Subject Headings (MeSH) thesaurus, which is a database of hierarchically-organized medical vocabulary [24]. The Unified Medical Language System (UMLS) [18] integrates key international biomedical terminology, classification, and coding standards and tools to promote interoperability among health information systems. The specialized knowledge available in these resources could aid NLP models in learning the associations between distinctive terminologies and better understanding the input text.

Biomedical Contextual Models

Naturally, there is a strong interest in NLP models among health data scientists. Direct application of the contextual models described in the previous sections usually falls short in the biomedical domain, because the distinctive terminologies and idioms are not always present in publicly available training datasets [58]. Thus, many researchers have focused on creating contextual models specially tailored for the biomedical domain. BioBERT [58] was trained on PubMed abstracts and PubMed Central full-text articles. Experiments on the BioBERT model demonstrated that incorporating biomedical corpora in the pre-training process could improve the model's performance on different downstream biomedical tasks [58]. Bio_ClinicalBERT [4] and BlueBERT [81] were further trained on clinical notes (e.g., the Medical Information Mart for Intensive Care (MIMIC) III dataset [47]) to improve their performance on clinical-related downstream tasks. Beltagy et al. introduced SciBERT [14] a contextual model trained in different research papers in both biomedical and computer science domains.

He et al. [37] inserted disease knowledge into existing models by training them to predict disease names and aspects (e.g., symptoms, diagnosis and treatment) based on Wikipedia passages. Similarly, Hao et al. [35] introduced a new pre-trained task to enable a BERT-based model to infer the existence of a relation between two medical concepts. These strategies have been shown to have a positive effect on model performance in multiple medical downstream tasks (i.e., entity recognition and natural language inference).

However, current biomedical applications of transformer-based Natural Language Processing models do not incorporate structured medical knowledge from a standard knowledge base (e.g., the UMLS [18] Metathesaurus) into their architecture. By integrating structured medical domain knowledge, a model would more easily learn the associations between distinctive terminologies, which it otherwise would not have the opportunity to learn due to the scarcity of medical datasets.

Contextual Model for Medical Conversation Summarization

Another promising application of medical knowledge-augmented contextual models is in the medical conversation summarization.

The documentation requirements for electronic health records (EHR) is a significant factor contributing to physician burnout [112, 107]. Various solutions have been proposed for automatically creating medical documentation to reduce documentation workloads, such as automatic speech recognition for dictating medical documents and medical notes generation. Studies have shown that these solutions significantly improve the efficiency of physicians in creating narrative reports [80].

Medical note generation by abstractive summarization can be used to automate clinical documentation to reduce the workload associated with creating summaries of clinical encounters. The model can take a transcript of a patient-doctor conversation as input and automatically produces a summary of the relevant clinical discussion in the dialogue [29].

There have been many attempts at developing automatically generated summaries of clinical encounters to date. Most notably, Enarvi et al. [27] proposed a seq-to-seq pointer generator transformer model for summarizing doctor-patient conversations. Similarly, Jeblee et al. [45] and Lacson et al. [56] used extractive methods to identify the most important utterances from a conversation, which were then combined to form the final summary.

However, these models have yet to take advantage of structured medical information, which could help key information pass the model's decision process and appear in the summary. Furthermore, one of the main challenges facing the development of medical summarization models is the lack of large-scale annotated summarization datasets. Their creation requires trained doctors for an expensive and time-consuming annotation process. Thus, a knowledge-augmented sequence-to-sequence transformer model that uses medical knowledge can guide the summarization process in various ways to increase the likelihood of relevant medical facts being included in the summarized output.

Contextual Models for Imbalanced Multi-Label Classification Problems in the Biomedical Domain

Another important biomedical application of transformer-based Natural Language Processing models is the automatic ICD coding problem, which is a highly imbalanced multi-label classification problem.

The International Classification of Diseases (ICD) is a widely used coding system, maintained by the World Health Organization [8]. The ICD arranges the codes hierarchically from general to more specific codes that are accompanied by non-essential modifiers. Assigning the most appropriate codes is an important task in healthcare, since erroneous ICD codes could seriously affect the organization's ability to measure patient health outcomes.

Recent attempts at using contextual models on the ICD classification task have failed to achieve state-of-the-art results [125], mainly due to their inability to process long documents (e.g., medical notes). Fortunately, advances such as the BigBird model [122] allow contextual models to process long documents, thus reducing the risk of losing information from the original texts.

1.3 Problem Statement

As outlined in this chapter, previous works have shown that injecting structured domain-specific knowledge into contextual models can further improve their performance and efficiency. This could be a solution to the challenges facing NLP tasks that lack useful training resources. In this thesis, we hypothesize that incorporating lexical and semantic knowledge will significantly enhance the performance of transformer-based NLP models. The underlying motive of this inquiry seeks to answer the question: "Can structured domain-specific knowledge be effectively incorporated into contextual models to tackle the low-resource NLP challenge by forcing the model to learn the associations between distinctive terminologies which it otherwise may not have the opportunity to learn, due to the scarcity of domain-specific datasets, in particular in the biomedical domain?"

More specifically, we investigate the following research questions:

- Can lexical resources aid a contextual model in distinguishing which words are semantically similar?
- What is the best way for a contextual model to quickly and effectively learn the associations between distinctive terminologies from an external structured knowledge base?
- Can structured medical knowledge that is integrated into a sequence-to-sequence contextual model guide the summarization process to include relevant medical fact in the summarized output?
- Can the relations between different class labels be efficiently encoded to improve classification performance?

1.4 Contributions

The thesis illustrates how structured external knowledge can direct an NLP model to learn the associations between distinctive terminologies, which otherwise would have been impossible to identify due to the scarcity of domain-specific datasets. The impact of different strategies on improving the performance and generalization of contextual models is also examined. We develop several novel strategies for augmenting contextual models with structured domain knowledge and experimentally verified their impact on the model performance in different NLP tasks (Figure 1.4).

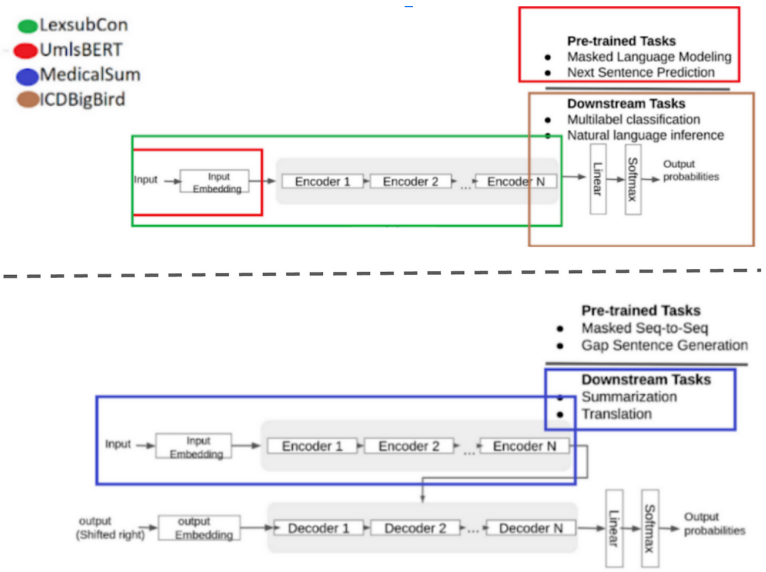


Figure 1.4: The main contribution of this dissertation

The contribution of this dissertation is summarized in the following four distinct but complementary strategies:

- We demonstrate that structured knowledge from a lexical database can aid a contextual model in distinguishing which words are semantically similar by developing a framework for incorporating general lexical knowledge, such as WordNet, into transformer models for the lexical substitution task. In particular, we design a new mix-up embedding strategy for the input embedding of the target word and introduce additional information, such as the degree of similarity between the definitions of the target and the candidate words.

- We confirm that augmenting a transformer-based encoder model with structured knowledge from a specific (medical) domain can aid the model in learning more easily the associations between distinctive terminologies by narrowing down the (general domain) lexical features to specific (medical) domain features. We introduce the usage of domain (medical) knowledge from a clinical Metathesaurus (UMLS Metathesaurus) in the pre-training phase of a BERT-based model (UmlsBERT) to build 'semantically enriched' contextual representations that will benefit from both the contextual learning (BERT architecture) and the domain knowledge (UMLS Metathesaurus);

- We demonstrate that a sequence-to-sequence summarization model which uses medical structured knowledge can guide the summarization process to include relevant medical facts in the summarized output by extending the medical features for augmenting a transformer-based encoder model with clinical knowledge. We answer the question of how to incorporate structured medical knowledge into a medical summarization model by designing specific 'guidance' signals over medical entities; and

- We show that a contextual model can also take advantage of the relational information between different labels by developing an attention mechanism for integrating a BigBird contextual model with information from the relation of different labels for a multi-label classification problem.

1.5 Outline

This dissertation is organized as follows:

- In Chapter 2, we propose an augmented contextual framework for the lexical substitution task. We show that integrating lexical structure information into a contextual model can improve the model's performance, as it outperforms other state-of-the-art models on two benchmark datasets. The results of the experiments and our qualitative analysis confirm that the additional information provided to our model, such as the degree of similarity between the definitions of the target and candidate words, can aid our model in selecting candidate words that are semantically similar to the target word rather than those that are only appropriate for the sentence's context.

- In Chapter 3, we present a novel architecture, namely UmlsBERT, for augmenting contextual embeddings with the Unified Medical Language Systems (UMLS) [18] by narrowing down from the general domain lexical features to medical domain-specific

features. We demonstrate that a transformer model which uses medical knowledge in its pre-training phase and its architecture can outperform two popular medical BERT models (i.e., BioBERT and Bio ClinicalBERT) and a general domain BERT model in different medical named-entity recognition (NER) tasks and one clinical natural language inference task. We also conduct a qualitative analysis which confirms that a model augmented with structured medical domain knowledge can learn effectively the associations between distinctive terminologies.

- In Chapter 4, we demonstrate that medical structured knowledge can also boost the performance of a transformer-based sequence-to-sequence model to summarize medical conversations by extending the medical features which were previously proposed to augment a transformer-based encoder model. We show that providing 'guidance' to a summarization model is beneficial for its performance, as it outperforms previous medical note summarization models. The results of these experiments and our qualitative analysis also demonstrate that these features can guide the summarization process and can increase the likelihood of relevant medical facts being included in the summarized output.

- In Chapter 5, we further examine whether a model can take advantage of both the relational information between different labels and contextual embedding information. This is accomplished through a novel attention mechanism, instead of augmenting the architecture of contextual models with structured information as described in the previous chapters, on a multi-label classification task, namely the ICD automatic coding problem. Our experiments verify that integrating relational information from the labels can be beneficial for the performance of a classification model as our model outperforms other state-of-the-art models on the MIMIC III benchmark dataset.

- The conclusion and future work are presented in Chapter 6. We describe our plan to explore a few promising avenues to further improve knowledge argumentation in contextual models.

Chapter 2

Augmenting Contextual Models with General Lexical Knowledge

2.1 Introduction

In Chapter 1, we described the main focus of this dissertation, which is to demonstrate that structured external knowledge can force an NLP model to learn associations between words that otherwise the model would not have the opportunity to learn. In this chapter, we will examine the benefit of incorporating lexical knowledge into a contextual model for the task of generating appropriate words which can replace a target word in a given sentence without changing the sentence's meaning (i.e., the lexical substitution task [66]). To demonstrate how this can be achieved, we develop LexSubCon, an end-to-end lexical substitution framework based on contextual embedding models that uses external structured knowledge to identify highly-accurate substitute candidates.

We will examine whether general-domain lexical resources can aid a contextual model in distinguishing which words are semantically similar. Our focus will be on the lexical substitution task [66] as a highly-accurate lexical substitution model can be utilized in various NLP fields, including data augmentation [13] and paraphrase generation [126].

Our first step will be to investigate whether changing the input policy of a contextual model to a mix-up scenario by linearly interpolating the target input embedding and the average embedding of its synonyms can boost the model's performance. We will examine whether incorporating information such as the similarity of the glosses (i.e., dictionary-style definition) of the target and the candidate words and the similarity of the initial and the updated sentences can aid the model in providing more accurate candidates.

We will also explore the effect of these features on two lexical substitution tasks:

1. the all-ranking task, where the model needs to identify and appropriately rank the potential candidate words;
2. the candidate ranking task, where a list of candidates is provided and the goal is to rank all the candidate words.

We will carry out a qualitative substitution comparison to show different cases where a contextual model can provide more accurate predictions by benefiting from information gathered from external resources.

The remainder of this chapter is organized into five parts. Section 2.2 presents related work, followed by details of the characteristics and individual features that are part of the proposed LexSubCon framework in Section 2.3. The experimental setup and data that are used to train and test the lexical substitution model are described in Section 2.4. The results of the experiments and the qualitative analysis are reported in Section 2.5 and Section 2.6 concludes the chapter.

2.2 Related Work

The lexical substitution task consists of two sub-tasks:

1. generating a set of meaning preserving substitute candidates for the target word;
2. appropriately ranking the words of the set by their ability to preserve the meaning of the initial sentence [32, 65].

However, lexical substitution models can also be tested in a 'simpler' problem where the set of substitute candidates is composed of human-suggested words and the task is to accurately rank the substitute words that are provided [28].

Melamud et al. [71] proposed the use of a word2vec model to rank the candidate substitutions by measuring their embedding similarity. Word2vec [72] was a popular word-embedding approach, representing each word on a fixed-size vector space through a hidden layer neural network which effectively captured semantic and syntactic word similarities. Later on, Roller and Erk [89] improved this approach by switching to a dot product instead of cosine similarity and applying an additional trainable transformation of the context

word embeddings. However, these models used a constant context-independent vector representation for each word. Therefore, they could not distinguish between a word's different meanings in a given context.

As introduced in Chapter 1, Transformer models [113] used self-attention layers to take into consideration the other positions/words in the input sequence while encoding a particular word. As such, they have become a key component in recent language models. The Bidirectional Encoder Representations from Transformers (BERT) [23] model took advantage of the Transformer's architecture by using its encoder component. The key idea behind BERT was that by utilizing the bidirectional training of Transformers for language modeling, the model could gain a better sense of the context. Thus, BERT achieved the state-of-the-art for major NLP tasks, including language inference [117] and text classification [115].

Zhou et al. [127] achieved state-of-the-art results on the lexical substitution task using the standard BERT architecture [23]. This was accomplished by applying a dropout embedding policy to the target word embedding. They also integrated into the raking metric the similarity between the original contextualized representations of the context words and their representations after replacing the target with one of the possible substitutes to ensure minimal changes in the sentence's meaning.

However, these models did not consider incorporating structured knowledge from external lexical databases into their prediction process. External lexical resources could boost the model's performance by providing additional information, such as the definitions of the target and candidate words. Having access to the definition could aid the model in selecting substitution words that are semantically similar to the target word rather than just appropriate for the sentence's context. External lexical resources could also enrich the proposed candidate word list so that it is not limited to the vocabulary of the contextual model.

2.3 LexSubCon Framework

To enhance the performance of previous approaches we developed LexSubCon, which we demonstrate can successfully combine contextual information with knowledge from structured external lexical resources.

The architecture of LexSubCon is depicted in Figure 2.1. The key characteristic of LexSubCon is its capability to integrate different substitution criteria such as contextualized

representation, definition, and sentence similarity into a single framework to accurately identify suitable candidates for the target words in a specific context (i.e., sentence).

In subsection 2.3.1, we will describe the initial BERT-Lexical substitution model [127] and will outline our contribution to replacing the embedding dropout policy of the target word with a new mix-up embedding strategy. In subsection 2.3.2, we will discuss our policy regarding the incorporation of the degree of similarity between sentence-definition (gloss) embeddings into the ranking metric. Finally, in subsection 2.3.3, we will present our proposed strategy concerning the creation of a fine-tuned sentence similarity model for calculating the effect of each substitution on the semantics of the sentence.

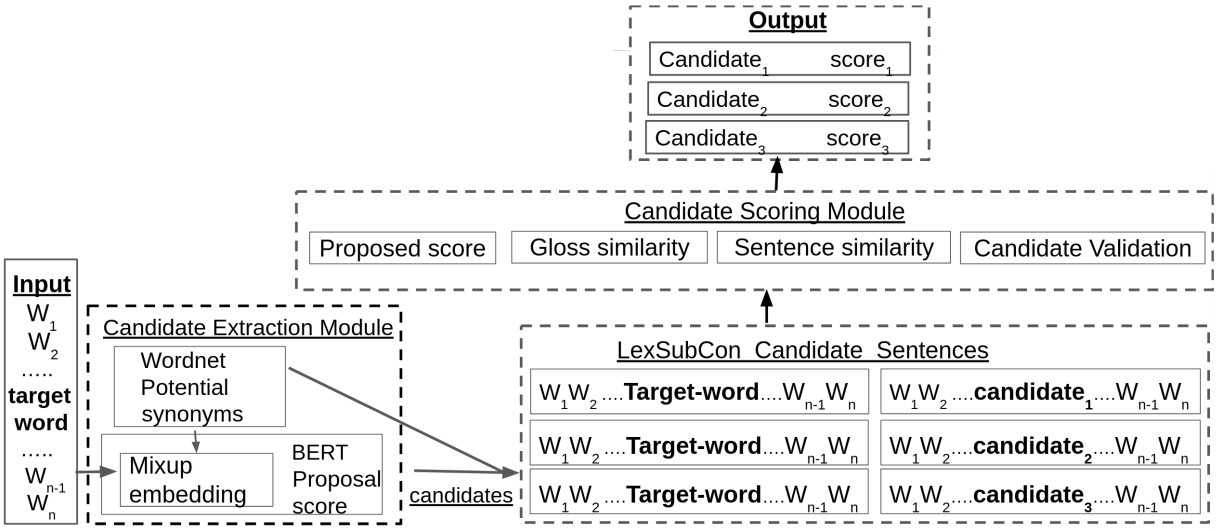


Figure 2.1: LexSubCon framework. LexSubCon proposes candidates for each target word by using external lexical resources and the BERT-based lexical substitution approach. It also ranks each candidate by considering different substitution criteria such as contextualized representation, definition, and sentence similarity.

2.3.1 Proposed Score: Mix-Up Embedding Strategy

The original BERT model [23] is based on multi-layer bidirectional transformers [113] which generate contextualized word representations. Incorporating information from bidirectional representations allows the BERT model to capture more accurately the meaning of a word based on its surrounding context (i.e., sentence).

The standard BERT architecture [23] can be used in the lexical substitution task by masking the target word and letting the model propose appropriate substitute candidates that preserve the initial meaning of the sentence. Zhou et al. [127] showed that applying embedding dropout to partially mask the target word is a better alternative than completely masking, or not masking, the target word. This is because the model may generate candidates that are semantically different but appropriate for the context of the initial sentence.

However, in this chapter, we will demonstrate that a mix-up embedding strategy can yield even better results. We propose that by using external knowledge, we can obtain probable synonyms of the target word and use that knowledge in a mix-up scenario [123]. This is achieved by linearly interpolating the target input embedding and the average embedding of its synonyms. This allows the model to generate a new synthetic input embedding by re-positioning the target embedding around the neighborhood of the embedding of its synonyms. In order to obtain appropriate synonyms, we use WordNet [73] an extensive lexical database where words are grouped into sets of synonyms called synsets. Our experiments achieve the best performance when the list of synonyms is extracted from the complete set of synsets for each word. This also minimizes the chances of having a synonym set that only includes the target word itself.

Finally, we use a mix-up strategy to calculate a new input embedding for the target word X'_{target} as shown in equation 2.1:

$$X^{'}_{target} = \lambda X_{target} + (1 - \lambda)\overline{X}_{synonyms} \tag{2.1}$$

where X_{target} is the initial input embedding of the target word, $\overline{X}_{synonyms}$ is the average embedding of all the synonyms, and λ is a hyper-parameter value. It should be noted that WordNet does not contain information about pronouns, conjunctions, or nouns that are not common in the English vocabulary. To address this limitation, whenever a target word cannot be found in the WordNet database, we replace the mix-up strategy by injecting Gaussian noise into the input embedding of the target word. This produces a similar effect as the mix-up strategy since the target embedding is re-positioned around itself in the embedding space (equation 2.2):

$$X^{'}_{target} = X_{target} + e \tag{2.2}$$

where e is a Gaussian noise vector with components $e_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$.

After updating the input embedding of the target word, we use the standard BERT architecture to calculate the proposal score for each candidate. The input embedding

vectors pass through multiple attention-based transformer layers, and each layer produces a contextualized embedding of each token. For each target word x_t, the model outputs a score vector of $y_t \in \mathbb{R}^D$, where D is the length of the model's vocabulary. We calculate the proposal score s_p, for each candidate word x_c, as the probability for the BERT model to propose the word x_c over all the candidate words x'_c when the target word's sentence is provided as input to it (equation 2.3):

$$s_p(x_c) = \frac{exp(y_t[x_c])}{\sum_{x'_c} exp(y_t[x'_c])} \tag{2.3}$$

where s_p is also the first feature of the candidate score s_c for each substitution candidate x_c, and α is the weight of the proposal score (equation 2.4):

$$s_c = \alpha \cdot s_p \tag{2.4}$$

2.3.2 Gloss-Sentence Similarity Score

In the previous section, we analyzed our model, which ranks candidate substitute words by calculating their individual proposal scores. In this section, we present a new metric that ranks the candidate words by considering the gloss (i.e., a dictionary-style definition) of each word. By extracting information from the WordNet database, a list of potential glosses is created for each target or candidate word. We then determine the most appropriate gloss based on the word and its specific context (sentence) by taking advantage of recent fine-tuned contextual models that have achieved state-of-the-art results in the Word Sense Disambiguation (WSD) task [41]. As the glosses are sentences (i.e., sequence of words), they can be represented on a semantic space through a sentence embedding generating model. Each candidate word is ranked by calculating the cosine similarity between the gloss sentence embedding of the target word and the gloss sentence embedding of each candidate word.

It should be noted that there are several methods for generating sentence embeddings, such as by calculating the weighted average of its word embeddings [6]. We decide to utilize the sentence embedding stsb-roberta-large model of Reimers et al. [87] which has been shown to outperform other state-of-the-art sentence embeddings methods.

Given a sentence s, a target word x_t, and a candidate word x_c, our model first identifies the most appropriate gloss g_t, for the target word given its context (by utilizing the pre-trained GlossBERT model [41] which has achieved state-of-the-art results in multiple

WSD tasks). After replacing the target word with the candidate x_c, to create a new sentence s', the most appropriate gloss g_c, for the candidate word is also determined (by also taking advantage of the pre-trained GlossBERT model[41]). A gloss-similarity score s_g, for each candidate is then calculated as the cosine similarity between the two glosses-sentence embeddings (equation 2.5).

$$s_g(x_c) = cos(g_t, g_c) \tag{2.5}$$

Finally, based on the work by Loureiro et al. [62] and our experiments, we found that uniting the synset lemma (i.e., a canonical form of a word) alongside the sentence of each gloss can have a beneficial effect on the comparison of the sentence embeddings, especially for the glosses that have limited length.

By calculating the gloss-similarity score, s_g, the candidate score for each candidate word, x_c, is updated to (equation 2.6):

$$s_c = \alpha \cdot s_p + \beta \cdot s_g \tag{2.6}$$

where β is the weight of the gloss similarity score.

2.3.3 Sentence Similarity Score

We also choose to assess the effect of each substitution on the semantics of the original sentence by calculating the semantic textual similarity between the original sentence (s) and an updated one (s'). An updated sentence is a sentence where we have replaced the target word with one of its substitutions.

Many pre-trained models for semantic textual similarity have become publicly available as described by Reimers et al. [87]. These models can be used in our task to measure the similarity of the initial sentence to each updated sentence after replacing the target word with one of the possible substitutes. However, to accurately calculate a similarity score between s and s', we first need to fine-tune the semantic textual similarity model, namely the stsb-roberta-large model [87]. This is achieved by using the training portion of the dataset to create pairs of sentences between the original sentence and an updated sentence where we have substituted the target word with one of its proposed candidates. In addition, using the methods described in subsection 2.3.2, we can identify the most appropriate synset (from WordNet) for each target word. We can then create a new pair of sentences between the original sentence and an updated sentence, where we have updated

the target word with the synonyms of the previously mentioned synset. However, due to the limited training dataset size, our model is still not provided with enough training data to be fully fine-tuned.

To remedy this situation, we employ a data augmentation technique to produce the examples needed for this task. Specifically, we create a back-translation mechanism to generate artificial training data. Back-translation or round-trip translation is the process of translating the text into another language (i.e., forward translation) and then translating it back into the original language (i.e., back-translation) [3]. Back-translation has been used in different tasks to increase the size of training data [95, 7]. In our case, we provide the initial sentence s to the back-translation module, which produces a slightly different 'updated' sentence s'_u. For the s'_u sentences that still contain the target word, we can create a pair of sentences between the s'_u and an alternative version of the s'_u sentence (s''_u) where the target word is substituted with one of the candidate words or synonyms that we mentioned in the above paragraph. The main disadvantage of this technique is that it may return the same initial sentence without any changes. In this case, a second translation level is added, where the initial sentence is translated into two different languages before being translated back.

After training our similarity model, we calculate the semantic textual similarity s_t, between the original and the updated sentence. Thus, the candidate score for each substitution candidate, x_c, can be updated to (equation 2.7):

$$s_c = \alpha \cdot s_p + \beta \cdot s_g + \gamma \cdot s_t \tag{2.7}$$

where γ is the weight of the sentence similarity score.

2.3.4 Candidate Validation Score

In our experiments, we have also include the substitute candidate validation metric from Zhou et al. [127] since it has been demonstrated to have a positive effect on the performance of a lexical substitution model. The substitute candidate validation metric is derived as the weighted sum of the cosine similarities between the contextual representation of each token in the initial sentence and in the updated one. The weight of the token, i, is calculated as the average self-attention score of all heads in all layers from the token of the target word to token i. According to Zhou et al. [127], this metric evaluates the influence of the substitution on the semantics of the sentence.

After the inclusion of the candidate validation metric s_v, the candidate score can be updated to (equation 2.8):

$$s_c = \alpha \cdot s_p + \beta \cdot s_g + \gamma \cdot s_t + \delta \cdot s_v \tag{2.8}$$

where δ is the weight of the validation score.

2.3.5 Candidate Extraction

The candidates for each target word are extracted using the external lexical resource of WordNet and the BERT-based lexical substitution approach, where the model provides probabilities for each candidate based on the context (i.e. sentence). We create a list of candidates based on the synonyms, hypernyms and hyponyms of each target word that could be identified in WordNet. The list also comprises the candidate words with the highest probabilities that could be identified using the BERT model and the mix-up strategy described in subsection 2.3.1. We choose to include candidates from WordNet because we do not want our model to be confined to candidate words from the BERT vocabulary alone. We also include candidate words from a BERT-based model because target words may not be included in WordNet or the lexical resource may only return the target word as a candidate.

2.4 Experiments

2.4.1 Dataset

We evaluate LexSubCon on the English datasets SemEval 2007 (LS07)[1] [66] and Concepts In-Context (CoInCo)[2] [54] which are the most widely used datasets for the evaluation of lexical substitution models.

1. The LS07 dataset is split into 300 training and 1710 testing sentences where for each of the 201 target words, there are 10 sentences [3]. The gold standard is based on manual annotation, where annotators provided up to 3 possible substitutes.

2. The CoInCo dataset consists of over 15K target word instances (based on texts from the Open American National Corpus), where 35% are training and 65% are testing data. Each annotator provided at least 6 substitutes for each target word.

[1] license: https://tinyurl.com/semeval-license
[2] license: CC-BY-3.0-US
[3] extracted from http://corpus.leeds.ac.uk/internet.html

In order to have a fair comparison with previous state-of-the-art models, we use processed versions for both datasets as used in [70, 71].

2.4.2 Experimental Setup

LexSubCon is evaluated in the following variations of the lexical substitution tasks:

All-ranking task: In this task, no substitution candidates are provided. We use the official metrics that the organizers provided in the original lexical substitution task of SemEval-2007[4]. These are *best* and *best-mode* which validate the quality of the model's best prediction and both *oot* (out-of-ten) and *oot-mode* to evaluate the coverage of the gold substitute candidate list by the 10-top predictions. We also use *Precision*@1 to have a complete comparison with the model in [127]. We use these metrics to evaluate our LexSubCon's substitution candidates in both the LS07 and CoInCo datasets.

Candidate ranking task: In this task, the list of candidates is provided, and the goal of the model is to rank all the candidate words. For the candidate ranking task, we follow the policy of previous works and construct the candidate list by merging all the substitutions of the target lemma and POS tag over the whole dataset. For measuring the performance of the model we use the GAP score [103][5] which is a variant of the Mean Average Precision (MAP). The generalized average precision (GAP) is calculated as (equation 2.9):

$$GAP = \frac{\sum_{i=1}^{n} I(x'_i)p_i}{\sum_{i=1}^{R} I(y_i)\overline{y_i}} \qquad p_i = \frac{\sum_{k=1}^{i} x_i}{i} \tag{2.9}$$

where x_i is a binary variable indicating whether the ith item provided by the model is in the gold standard or not and x'_i is the gold standard weight of the ith item or zero if the item is not in the gold standard. We define $I(x'_i) = 1$ if the x'_i is larger than zero and zero otherwise and $\overline{y_i}$ is the average weight of the ideal ranked list of gold standard y_1,...,y_i [103]. Following Melamud et al. [71], we discard all multi-words from the gold substitutes list and remove the instances that are left with no gold substitutes.

We use the uncased BERT large model [23] for calculating the proposal score and candidate validation score. For identifying the most appropriate glosses for the target word and its candidate, we employ the pre-trained model proposed by Huang et al. [41] which has achieved state-of-the-art results in multiple Word Sense Disambiguation (WSD)

[4] www.dianamccarthy.co.uk/files/task10data.tar.gz
[5] https://tinyurl.com/gap-measure

tasks and was trained on the SemCor3.0 dataset, the largest corpus manually annotated with WordNet sense for WSD [41]. The sentence-similarity metric is computed by fine-tuning the stsb-roberta-large model presented by Reimers et al. [87] and by employing the OPUS-MT models by Tiedemann, and Thottingal [106] (namely, opus-mt-en-romance, opus-mt-fr-es, and opus-mt-romance-en) for creating the back-translated sentences.

To address the reproducibility concerns of the NLP community [25] we provide the search strategy and the bound for each hyperparameter. We use the LS07 trial set for training the sentence similarity metric model (for 4 epochs) and for fine-tuning the parameters of our framework based on the *best* score. Empirically, the λ parameter of the mix-up strategy are set to 0.25 and the proposal score, gloss-sentence similarity score, sentence similarity score, and candidate validation score weights to $0.05, 0.05, 1, 0.5$, respectively (with the search space for all the parameters being $[0, 1]$)[6]. For the Gaussian noise, we select a mean value of 0 and a standard deviation of 0.01. We propose 30 candidates for each target word in each test instance. We run LexSubCon on five different (random) seeds to achieve more robust results and provide the average scores and standard deviation. All the contextual models are implemented using the transformers library [118] on PyTorch 1.7.1. All experiments are executed on a Tesla K80 GPU with 64 GB of system RAM on Ubuntu 18.04.5 LTS. LexSubCon contains 1136209468 parameters.

2.5 Results

2.5.1 Lexical Substitution Model Comparison

To enable direct comparison and to isolate gains solely due to improvements in the post-processing strategy that each model uses (which has the potential to change its performance [5]), we opt to reproduce and use the same strategy for the tokenization of the target words from Bert_{s_p,s_u} [127]. We focus our comparison on Bert_{s_p,s_u} as it has achieved impressive state-of-the-art results on both benchmark datasets[7].

The results of LexSubCon and the previous state-of-the-art results in both LS07 and CoInCo benchmark datasets are presented in Table 2.1. LexSubCon outperforms the previous methods across all metrics in the LS07 and the CoInCo datasets, given that all features

[6] As we only had four weight parameters, the identification of the best combination is finished in less than half an hour.

[7] Note that the method proposed in [127] is implemented as faithfully as possible, to the best of our abilities, to the original work, using elements of code kindly provided by the authors upon request. However, the authors could not make the complete original code available to us.

Method	best	best-m	oot	oot-m	P@1
LS07 dataset					
LexSubCon	**21.1** ±0.03	**35.5**±0.07	**51.3** ±0.05	**68.6**±0.05	**51.7**±0.03
Bert$_{s_p,s_u}$*	12.8 ±0.02	22.1±0.03	43.9±0.01	59.7±0.02	31.7± 0.02
T. L.	17.2	-	48.8	-	-
S. V.	12.7	21.7	36.4	52.0	-
Addcos	8.1	13.4	27.4	39.1	-
S. L.	15.9	-	48.8	-	40.8
UNT	12.8	20.7	49.2	66.3	-
CoInCo dataset					
LexSubCon	**14.0** ± 0.02	**29.7** ± 0.03	**38.0** ± 0.03	**59.2** ± 0.04	**50.5** ± 0.02
Bert$_{s_p,s_u}$ *	11.8 ± 0.02	24.2 ± 0.02	36.0 ± 0.02	56.8 ± 0.02	43.5 ± 0.02
S. V.	8.1	17.4	26.7	46.2	-
Addcos	5.6	11.9	20.0	33.8	-

Table 2.1: Results of mean ± standard deviation of five runs from our implementation of LexSubCon and Bert$_{s_p,s_u}$*[127]. We also provide the performance of previous state-of-the-art models. Transfer learning (T. L.) [39], Substitute vector (S. V.) [69], Addcos [71], Supervised learning (S. L.) [102], UNT [36]. Best values are **bolded**.

have a positive effect on its performance (see ablation details in subsection 2.5.2). This is because the features encourage LexSubCon to take into consideration different substitution criteria such as contextualized representation, definition, and sentence similarity. The standard deviation of the results of LexSubCon is not zero due to the fine-tuning process of the sentence similarity model. However, the results indicate that there are no large fluctuations.

2.5.2 Ablation Study

In order to evaluate the effect of each feature on the performance of LexSubCon, we conduct an ablation study. The results are presented in Table 2.2. LexSubCon achieves its best performance when it has access to information from all the features described in Section 2.3 (first row in Table 2.2). By testing the performance of the individual features, we observe that the gloss sentence similarity feature results in the worst performance out of all the features. This is likely because many candidate words cannot be identified in WordNet and thus we assign a zero value to their gloss sentence score. Another factor is that the models used to select the most appropriate gloss for each word may introduce noise in the

Method	best	best-m	oot	oot-m	*P*@1
LS07					
LexS	**21.1**	**35.5**	**51.3**	**68.6**	**51.7**
-w Pr.	20.1	32.6	50.8	68.1	50.6
-w Gl.	19.9	33.7	50.4	67.6	48.6
-w Sen.	20.7	34.9	50.9	68.2	50.6
-w Val.	18.8	31.7	47.8	64.9	46.6
Pr.	16.3	27.6	45.6	62.4	40.8
Gl.	12.4	19.5	40.5	55.0	32.7
Sen.	16.7	28.3	45.3	62.0	40.7
Val.	18.6	30.8	48.9	66.2	46.3
CoInCo					
LexS	**14.0**	**29.7**	**38.0**	**59.2**	**50.5**
-w Pr.	12.9	26.5	37.6	58.5	47.8
-w Gl.	13.4	28.5	37.2	58.2	48.8
-w Sen.	13.6	29.9	37.2	58.3	49.2
-w Val.	12.7	27.0	35.9	57.4	46.6
Pr.	11.3	23.8	33.6	54.4	41.3
Gl.	8.4	16.7	29.6	47.2	33.6
Sen.	10.9	22.5	34.0	54.9	40.5
Val.	11.7	23.7	35.3	55.2	44.2

Table 2.2: Ablation study of LexSubCon: *Pr.* is the Proposal score using the mix-up embedding strategy. *Gl.* is the Gloss similarity score. *Sen.* is the Sentence Similarity score and *Val.* is the Validation score. *-w/o* indicates a LexSubCon framework **without** the specific feature.

process of the gloss-similarity score as they may select non-optimal glosses.

2.5.3 Mix-Up Strategy Evaluation

In order to evaluate the mix-up strategy, we study the effect of different input embedding policies. The results are shown in Table 2.3. Even the simpler strategy of injecting Gaussian noise into the input embedding outperforms the standard policy of masking the input word. These results indicate that a contextual model needs information from the embedding of the target word to predict accurate candidates.

However, the model may over-rely on this information when provided with an intact

Policy	best	best-m	oot	oot-m	*P*@1
LS07					
Mix.	**16.3**	**27.6**	**45.6**	**62.4**	**40.8**
Gaus.	15.4	25.1	44.3	61.4	38.9
Drop.	15.5	25.6	44.3	61.2	38.8
Mask	10.4	16.4	35.5	48.6	27.0
Keep	15.5	25.4	44.4	61.4	39.2
CoInCo					
Mix.	**11.3**	**23.8**	**33.6**	**54.4**	**41.3**
Gaus.	10.8	22.6	33.0	54.4	39.7
Drop.	10.8	22.5	32.9	54.2	39.5
Mask	8.6	17.5	28.9	46.6	31.7
Keep	10.8	22.6	33.0	54.3	39.7

Table 2.3: Comparison of different strategies for modifying the input embedding. *Mix.* is the mix-up strategy that we proposed, *Gaus.* is the Gaussian noise strategy, *Drop.* is the dropout embedding strategy [127], *Mask* is the strategy of masking the target word and *Keep* is the strategy of unmasking the target word. Best values are **bolded**.

input embedding. The mix-up strategy outperforms all the other policies, specifically the dropout embedding strategy [127]. This is because the mix-up strategy re-positions the target embedding around the neighborhood of the embedding of its synonyms, so it does not erase a part of the embedding that the model can learn from.

2.5.4 Candidate Ranking Task

We also evaluate LexSubCon in the candidate ranking task for both the LS07 and CoInCo datasets. As mentioned in Section 2.4.2, in this task the candidate substitution words are provided. The main goal is to create the most appropriate ranking of the candidates for each test instance.

Table 2.4 reports the evaluation results from the candidate ranking task of LexSubCon, as well as the results from the previous state-of-the-art models. As it can be observed, all the features positively affect the performance of LexSubCon, thus outperforming the previous state-of-the-art methods. The results demonstrate the features' positive effect on accurately ranking a list of potential candidates since the LexSubCon outperforms all the previous methods, even in the scenario where all the methods are provided with the same substitution candidate list.

Method	LS07	CoInCo
LexSubCon	**60.6**	**58.0**
-w/o Pr.	58.8	56.3
-w/o Gl.	60.3	57.4
-w/o Sen.	59.8	57.1
-w/o Val.	56.8	53.8
Bert$_{s_p,s_u}$*	58.6	55.2
LexSubCon (trial+test)	60.3	58.0
Bert$_{s_p,s_u}$* (trial+test)	57.9	55.5
XLNet+embs	57.3	54.8
context2vec	56.0	47.9
Trans. learning	51.9	-
Sup. learning	55.0	-
PIC	52.4	48.3
Substitute vector	55.1	50.2
Addcos	52.9	48.3
Vect. space mod.	52.5	47.8

Table 2.4: Comparison of GAP scores (%) from previously published results in the candidate ranking task of our implementation of LexSubCon and Bert$_{s_p,s_u}$ [127]. We also provide the results on the entire dataset with (trial+test). Models: XLNet+embs [5], Context2vec [70], Transfer learning [39], Supervised learning[102], PIC [90], Substitute vector [69], Addcos [71] and Vector space modeling [54].

2.5.5 Qualitative Substitution Comparison

Word	Sentence	Gold Ranking	LexSubCon	BERT$_{\text{based}}$
terrible	..have a **terrible** effect on the economy	awful, very bad, appalling, negative, formidable	horrible, horrific, awful	negative, major, positive
return	..has been allowed to **return** to its wild state	go back, revert, resume, regress	revert, retrovert, regress	recover, go, restore

Table 2.5: Examples of target words and their top lexical substitutes proposed by LexSubCon and BERT$_{\text{based}}$ model.

Table 2.5 reports different examples of target words and their top lexical substitutes

proposed by LexSubCon and the $\text{BERT}_{\text{based}}$ model in order to demonstrate the effect of external lexical resources on the performance of a contextual model. As it can be observed, for the target word *terrible*, the $\text{BERT}_{\text{based}}$ model proposes a candidate word (*positive*) that may fit in the sentence but has the opposite meaning of the target word. However, LexSubCon provides semantically similar candidates by using information from different signals (e.g., comparison of the definition of each word). For the target word '*return*', our model identifies an appropriate candidate that is not in the vocabulary of the contextual model (the word '*regress*') by introducing candidates from an external lexical database. These examples show that a contextual model enriched with external lexical knowledge can provide more accurate candidates.

2.5.6 Extrinsic Evaluation: Data Augmentation

Finally, we evaluate the performance of LexSubCon within the context of textual data augmentation. We conduct experiments using a popular English benchmark text classification task on a subjectivity/objectivity dataset (SUBJ) [79][8]. The SUBJ dataset contains 5000 subjective and 5000 objective processed sentences (based on movie reviews). We train the LSTM model (with the same hyperparameters) which was used in [116] to measure the effect of different data augmentation techniques. We then compare our method with two previous state-of-the-art lexical substitution models and other popular textual data augmentation techniques. These are:

1. the back-translation technique (described in Section 2.3.3); and
2. the EDA framework [116] which utilizes four operations of Synonym Replacement and Random Insertion/Swap/Deletion in order to create new text.

Following the data generation algorithm by Arefyev et al. [5], LexSubCon creates new examples by sampling one word for each sentence, generating the appropriate substitute list for this word, and sampling one substitute with probabilities corresponding to their substitute scores (which are normalized by dividing them by their sum) to replace the original word with the sampled substitute.

Figure 2.2 demonstrates how data augmentation affects the classification depending on the size of the training set [5, 116]. It is shown that the data created with lexical substitution has a more positive effect on the performance of the model compared to

[8]license: https://tinyurl.com/t-license

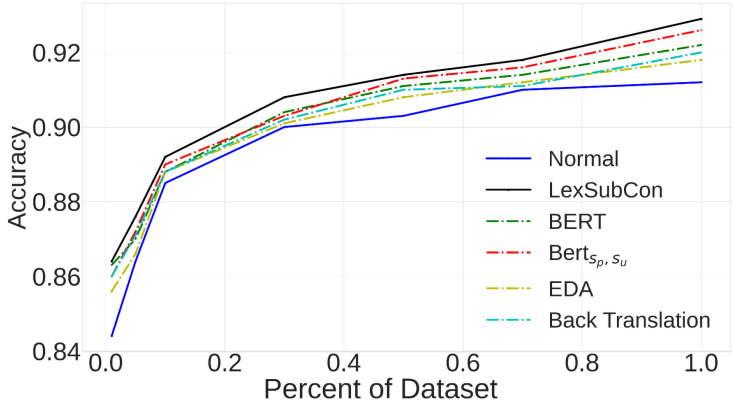


Figure 2.2: Accuracy with different training sizes for different text augmentation techniques on the SUBJ dataset.

other data augmentation techniques. This is likely because back-translation techniques may provide text that does not follow the syntactic rules of the target language. The EDA framework may also create examples that could confuse the model by changing the sentence structure due to the random insertion and swapping of words. Since LexSubCon creates more accurate substitution candidates than the standard BERT and the Bert$_{s_p,s_u}$* models, the texts created by LexSubCon have a more positive effect on the model's performance.

2.6 Conclusion

In this chapter we demonstrated that injecting external knowledge from a general lexical database into a contextual model can aid the model in distinguishing semantically similar words. Our model established a new mix-up embedding strategy that re-positioned the target embedding around the neighborhood of the embedding of its synonyms. Our model benefited from the combined usage of features from both the contextual embedding models and external lexical knowledge bases, such as a new gloss (definition) similarity metric, which could calculate the similarity of the sentence-definition embeddings of the target word and its proposed candidates. We also generated a highly accurate fine-tuned sentence similarity model by taking advantage of popular data augmentation techniques (such as back translation) to calculate each candidate word's effect on the semantics of the original sentence.

Our experiments showed that all features can aid the model in making accurate predic-

tions as LexSubCon achieved its best performance when it had access to all the features. LexSubCon outperformed previous state-of-the-art models by at least 2% over all the official lexical substitution metrics on LS07 and CoInCo benchmark datasets that are widely used for lexical substitution tasks. Finally, our qualitative analysis demonstrated that combining a contextual model with structured external knowledge can assist the model in selecting more accurate candidates.

Having successfully proven that general lexical structured knowledge can aid a contextual model in distinguishing between semantically similar words, we will extend this exploration in the following chapter to see if this approach can be adapted in the presence of highly technical or specialized vocabularies such as that found in medical text.

Chapter 3

Augmentation of Contextual Models in the Biomedical Domain

3.1 Introduction

In the previous chapter, we discussed how structured general lexical knowledge could boost the performance of a contextual model in the lexical substitution task by helping the model to distinguish between semantically similar words.

In this chapter, we will narrow down the general lexical features to specific-domain (medical) features, and we will investigate the effect of augmenting a transformer-based encoder model with structured knowledge from the medical domain (e.g., the UMLS metathesaurus [18]). Our proposed architecture will augment its input embedding layer with structured medical information, thus permitting it to consider the different semantic types of the medical words in the input sentence. Our model will also be trained with an 'updated' pre-trained task, enabling it to learn the connection of the medical words associated with the same concept in a medical metathesaurus.

While medical-focused contextual models already exist, we will demonstrate that our model is more suited for different medical downstream tasks. By integrating structured medical domain knowledge into a contextual model, we will show that the model can learn more easily the associations between distinctive terminologies, which it otherwise would not have the opportunity to learn due to the scarcity of medical datasets.

We chose to integrate medical information from the UMLS Metathesaurus as it is a compendium of many biomedical terminologies (e.g., MeSH [24] and ICD [76]) with their

associated information, such as synonyms and categorical grouping. It also allows for the connection of words that represent the same or similar 'concept'. For example, the words 'lungs' and 'pulmonary' share a similar meaning and thus can be mapped to the same concept unique identifier (CUI) *CUI: C0024109*. Additionally, UMLS allows the grouping of concepts according to their semantic type [67]. For example, 'skeleton' and 'skin' have the same 'Body System' semantic type, and 'inflammation' and 'bleed' are in the 'Pathologic Function' semantic type group.

The remainder of this chapter is organized into five parts. Section 3.2 provides an overview of related work, which is followed by a detailed account of the characteristics of the proposed UmlsBERT architecture for augmenting contextual embeddings with structured clinical knowledge in Section 3.3. Section 3.4 describes the experimental setup along with the data used to pre-train and test UmlsBERT. The results of the downstream tasks and the qualitative analysis are reported in Section 3.5, followed by the chapter conclusion in Section 3.6.

3.2 Related Work

3.2.1 BERT Model

The Bidirectional Encoder Representations from Transformers (BERT) [23] model achieved the state-of-the-art for major NLP tasks including language inference [117] and text classification [115] by utilizing bidirectional Transformers [113] to create context-dependent representations of the words in the input text. The pre-training of the BERT model was done on massive corpora, and the context-sensitive embeddings could be further fine-tuned for a downstream task by being integrated into a task-specific architecture.

The pre-training phase of the BERT model [23] consisted of two self-supervised tasks: (i) Masked Language Modelling (LM), in which a percentage of the input was masked at random and the model was forced to predict the masked tokens; and (ii) Next Sentence Prediction, in which the model had to determine whether two segments appear consecutively in the original text. Specifically, in the Next Sentence Prediction task, the model is provided with pairs of sentences with a 50% chance that the second sentence was actually the sentence that follows the first sentence [23].

In Masked LM, 15% of the tokens of each sentence were replaced by a [MASK] token. For the j^{th} input token in the sentence, an input embedding vector $u^{(j)}_{input}$ was created by the following equation (equation 3.1):

$$u_{input}^{(j)} = p^{(j)} + SEGseg_{id}^{(j)} + Ew_j \tag{3.1}$$

where $p^{(j)} \in \mathbb{R}^d$ was the position embedding of the j^{th} token in the sentence, and d was the transformer's hidden dimension. Additionally, $SEG \in \mathbb{R}^{d \times 2}$ was called the segment embedding, and $seg_{id} \in \mathbb{R}^2$, a 1-hot vector, was the segment id that indicates the sentence to which the token belongs. In Masked LM, the model used only one sentence, and therefore, the segment id indicated that all the tokens belong to the first sentence. $E \in \mathbb{R}^{d \times D}$ was the token embedding where D was the length of the model's vocabulary and $w_j \in \mathbb{R}^D$ was a 1-hot vector corresponding to the j^{th} input token.

The input embedding vectors passed through multiple attention-based transformer layers where each layer produced a contextualized embedding of each token. For each masked token w, the model output a score vector $y_w \in \mathbb{R}^D$ to minimize the cross-entropy loss between the softmax of y_w and the 1-hot vector corresponding to the masked token (h_w) (equation 3.2):

$$loss = -log(\frac{exp(y_w[w])}{\sum_{w'} exp(y_w[w'])}) \tag{3.2}$$

3.2.2 Biomedical Contextual Model

There have been multiple attempts to improve the performance of contextual models in the biomedical domain.

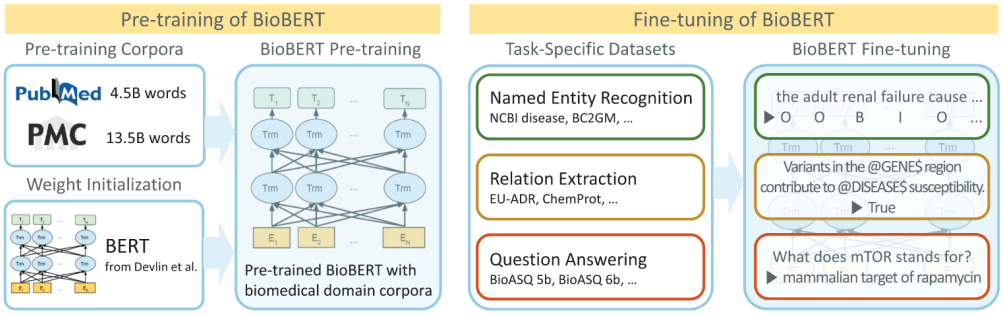


Figure 3.1: Overview of the pre-training and fine-tuning of BioBERT [58].

BioBERT was a BERT-based model which was pre-trained on both general (BooksCorpus and English Wikipedia) and biomedical corpora (PubMed abstracts and PubMed Central full-text articles) (Figure 3.1). BioBERT illustrated the positive effect of training a

contextual model with medical data, as it outperformed the standard BERT in multiple downstream tasks (e.g., named entity recognition and relation extraction). This was likely because medical corpora contains terms that were not usually found in a general domain corpus [34]. Bio_ClinicalBERT [4] further pre-trained BioBERT on clinical text from the MIMIC-III v1.4 database [47]. It was shown that further pre-training with clinical-specific text can be beneficial for the performance of a model on different clinical NLP downstream tasks [4]. Finally, PubMedBERT [33] was pretrained from scratch using abstracts from PubMed (which contains more than 34 million citations and abstracts of biomedical literature) and achieved state-of-the-art performance on several biomedical NLP tasks. However, the models mentioned above failed to take into account the relations between medical entities that exist in medical databases.

He et al. [37] infused disease knowledge into a BERT-based model by training the model to predict disease names and aspects on Wikipedia passages. Hao et al. [35] also introduced a new pre-trained task to enable a BERT-based model to infer the existence of a relation between two medical concepts. These strategies have been shown to positively affect the model's performance on multiple medical downstream tasks, e.g., entity recognition and natural language inference.

However, current biomedical applications of transformer-based Natural Language Processing models did not incorporate expert structured (medical) domain knowledge from a knowledge base (e.g., the UMLS [18] Metathesaurus) into their architecture. By integrating structured medical domain knowledge, a model would more easily learn the associations between distinctive terminologies.

To enhance the performance of previous approaches we developed UmlsBERT, which we demonstrate can successfully integrate (medical) domain knowledge during its pre-training process via a novel knowledge augmentation strategy.

3.3 Methods

In this section, we will present the proposed architecture for integrating UMLS-based features in the UmlsBERT's pre-training process and architecture. In particular, we will analyze the methodology for enriching input embeddings with semantic type information and present the new loss function, which is used to learn the connection of words through their corresponding CUI's.

3.3.1 Semantic Type Embeddings

We introduce a new embedding matrix called $ST \in \mathbb{R}^{D_s \times d}$ into the input embedding of the BERT model, where d is BERT's transformer hidden dimension and $D_s = 44$ is the number of unique UMLS semantic types that can be identified in the vocabulary of our model. In particular, in this matrix, each row represents the unique semantic type in UMLS that a word can be identified with (for example, the word 'heart' is associated with the semantic type T023:'Body Part, Organ, or Organ Component' in UMLS).

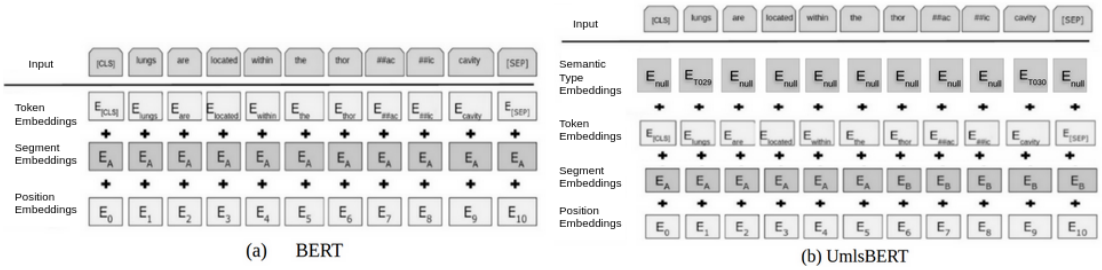


Figure 3.2: **(a)** Original input vector of the BERT model [23]. **(b)** Augmented input vector of the UmlsBERT where the semantic type embeddings is available. For the words 'lungs' and 'cavity', their word embeddings are enhanced with the embedding of the semantic type 'Body Part, Organ, or Organ Component'(E_{T023}) and 'Body Space or Junction'(E_{T030}) respectively. The rest of the words are not related to a medical term, so a zero-filled tensor E_{null} is used.

To incorporate the ST embedding matrix into the input embedding of our model, all words with a clinical meaning defined in UMLS are identified. The corresponding concept unique identifier (CUI) and semantic type are extracted for each of these words. We use $s_w \in \mathbb{R}^{D_s}$ as a 1-hot vector corresponding to the semantic type of the medical word w. The identification of the UMLS terms and their UMLS semantic type is accomplished using the open-source Apache clinical Text Analysis and Knowledge Extraction System (cTakes) [93]. It should be noted that we acknowledge that one limitation of our model is that relies on the cTakes tools and thus it will not take advantage of any medical information that the cTakes did not identify. Thus, by introducing the semantic type embedding, the input vector (equation 3.1) for each word is updated to (equation 3.3):

$$u_{input}^{(j)\prime} = u_{input}^{(j)} + ST^{\top} s_w \tag{3.3}$$

where the semantic type vector $ST^{\top} s_w$ is set to a zero-filled vector for words not identified in UMLS. We hypothesize that incorporating the clinical information of the

semantic types into the input tensor could be beneficial for the performance of the model as it can be used to enrich the input vector of words that are rare in the training corpus and the model does not have the chance to learn meaningful information for their representation. Figure 3.2 presents an example of inserting the semantic type embeddings into the standard BERT architecture.

3.3.2 Updating the Loss Function of Masked LM Task

We update the loss function of the Masked LM pre-training task to take into consideration the connection between words that share the same CUI. As described in subsection 3.2.1, the loss function of the Masked LM pre-training task of a BERT model is a cross-entropy loss between the softmax vector of the masked word and the 1-hot vector that indicates the actual masked word. We choose to 'soften' the loss function and update it to a multi-label scenario by using information from the CUIs.

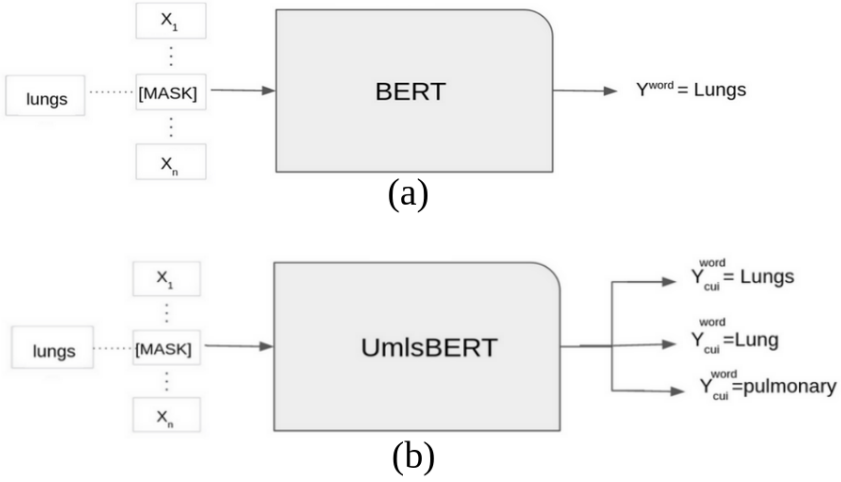


Figure 3.3: An example of predicting the masked word 'lungs' **(a)** the BERT model tries to predict only the word lungs, whereas **(b)** the UmlsBERT tries to identify all words that are associated with the same CUI (e.g lungs, lung, pulmonary).

More specifically, instead of using a 1-hot vector (h_w) that corresponds only to the masked word w, we use a binary vector indicating the presence of all the words which share the same CUI of the masked word (h'_w). In order for the model to properly function in a multi-label scenario, the cross-entropy loss (equation 3.2) is updated to a binary cross-entropy loss (equation 3.4):

$$loss = -\sum_{i=0}^{D}(h_w^{'}[i]log(y_w[i]) + (1 - h_w^{'}[i])log(1 - y_w[i])) \tag{3.4}$$

These changes force UmlsBERT to learn the underlying semantic relations between words associated with the same CUI in a biomedical context.

An example of predicting the masked word 'lungs' with and without the clinical information is presented in Figure 3.3. As seen in this figure, the UmlsBERT model tries to identify the words 'lung', 'lungs', and 'pulmonary', because all three words are associated with the same *CUI: C0024109* in the UMLS Metathesaurus.

3.4 Experiments

3.4.1 Dataset

We use the Multiparameter Intelligent Monitoring in Intensive Care III (MIMIC-III) dataset [47] to pre-train the UmlsBERT model. The MIMIC dataset consists of anonymized electronic medical records in English of over forty-thousand patients admitted to the intensive care units of the Beth Israel Deaconess Medical Center (Boston, MA, USA) between 2001 and 2012. In particular, UmlsBERT is trained on the **NOTEEVENTS** table, which contains 2,083,180 rows of clinical notes and test reports.

Dataset	Train	Dev	Test	C
MedNLi	11232	1395	14 22	3
i2b2 2006	44392	5547	18095	17
i2b2 2010	14504	1809	27624	7
i2b2 2012	6624	820	5664	13
i2b2 2014	45232	5648	32586	43

Table 3.1: Number of sentences for the train/dev/test set of each dataset. We also include the number of classes (C) for each dataset. We use the same splits that are used in the Bio_ClinicalBERT model [4].

We evaluate the effects of the novel features of the UmlsBERT model on the English MedNLI natural language inference task [91] and on four i2b2 NER tasks (in IOB format [86]). More specifically, we experiment on the following English i2b2 tasks: the i2b2 2006

de-identification challenge [110]; the i2b2 2010 concept extraction challenge [111]; the i2b2 2012 entity extraction challenge [99]; and the i2b2 2014 de-identification challenge [98]. These datasets are chosen because of their use in benchmarking prior biomedical BERT models, thereby allowing for performance comparison. In addition, these publicly available datasets enable the reproducibility of our results and allow for meaningful comparison with future studies. Table 3.1 lists the statistics of all the datasets.

3.4.2 UmlsBERT Training

We initialize UmlsBERT with the pre-trained Bio_ClinicalBERT model [4]. Since only the Masked LM task is affected by our modifications (i.e., the updated loss function) we omit the training of UmlsBERT on the NSP task as it will not meaningfully affect the performance of our model. Afterward, to perform the downstream tasks, we add a single linear layer on top of UmlsBERT and 'fine-tune' it to the task at hand, using either the associated embedding for each token or the embedding of the *[CLS]* token. The same fine-tuning method is applied to all other models used for comparison. To keep the experiment controlled, we use the same vocabulary, and WordPiece tokenization [119] across all the models.

In the pre-training phase, UmlsBERT is trained for $1,000,000$ steps with a batch size of 64, maximum sequence length of 128, and a learning rate of $5 \cdot 10^{-5}$. All other hyper-parameters are kept to their default values. UmlsBERT is trained by using 2 Nvidia V100 GPUs with 128 GB of system RAM running Ubuntu 18.04.3 LTS.

3.4.3 Hyperparameter Tuning

Our search strategy and the bound for each hyperparameter are: the batch size is set between 32 and 64, and the learning rate is chosen among the values 2e-5, 3e-5 and 5e-5. For the clinical NER tasks, we take a similar approach to the BioBert's experiments [58] and set the number of training epochs to 20 to allow for maximal performance (for the MedNLI task, we train the models on 3 and 4 epochs).

For the i2b2 tasks the best values are chosen based on validation set F1 values using the seqevals python framework for sequence labeling evaluation. This is due to the fact that it can provide an evaluation of a NER task on entity-level[1]. For the MedNLI task we choose the best values based on validation set accuracy, which is the standard metric

[1] https://github.com/chakki-works/seqeval

Dataset		$\text{BERT}_{\text{based}}$	BioBERT	Bio_ClinicalBERT	UmlsBERT
MedNLI	epochs	4	4	4	3
	batch size	16	16	32	16
	learning rate	5e-5	3e-5	3e-5	3e-5
i2b2 2006	epochs	20	20	20	20
	batch size	32	16	16	32
	learning rate	2e-5	2e-5	2e-5	5e-5
i2b2 2010	epochs	20	20	20	20
	batch size	16	32	32	16
	learning rate	3e-5	3e-5	5e-5	5e-5
i2b2 2012	epochs	20	20	20	20
	batch size	16	32	16	16
	learning rate	3e-5	3e-5	5e-5	5e-5
i2b2 2014	epochs	20	20	20	20
	batch size	16	16	32	16
	learning rate	2e-5	2e-5	5e-5	3e-5

Table 3.2: Hyperparameter selection of all the models for each dataset.

for this task [2]. To provide a fair comparison, we also tune the hyperparameters of each model to demonstrate its best performance. The final hyper-parameters selection of all the models for each dataset can be found in Table 3.2. It should be noted that, since BERT_{base}, BioBERT and Bio_ClinicalBERT use the same BERT-based architecture, they have the exact same number of parameters. However, because we introduce the semantic type embeddings into the UmlsBERT model, our model has an additional 33792 [the number of unique UMLS semantic types (44) $\times$ transformer's hidden dimension(768)] parameters[3]. Table 3.3 provides the number of parameters for each model where we also include the linear layer on top of the BERT-based models for each task.

3.5 Results

In this section, we present the results of an empirical evaluation of the UmlBERT model. In particular, we provide a comparison between different available BERT models to show

[2] https://tinyurl.com/transformers-metrics

[3] UmlsBert also contains an additional zero-filled vector, that we use, as the semantic type vector of the non-medical words, which is not included in the calculation of the number of the parameters.

the efficiency of our proposed model on different clinical NLP tasks. We also provide the results of an ablation test to examine the effect of semantic type embeddings on the model's performance. We conduct a qualitative embedding analysis to illustrate that our model can learn the association of different clinical terms with similar meaning in the UMLS Metathesaurus. Finally, we provide a visualized comparison of the embeddings of the words associated with semantic types between UmlsBERT and Bio_ClinicalBert to demonstrate the ability of our model to create more meaningful input embeddings.

3.5.1 Downstream Clinical NLP Tasks

BERT-based model comparison

In this section, we report the results of the comparison of our proposed UmlsBERT model with the other BERT-based models in different downstream clinical NLP tasks described in Section 3.4. All BERT-based models are implemented using the transformers library [118] on PyTorch 0.4.1. All experiments are executed on a Tesla P100 with 32G GB of system RAM on Ubuntu 18.04.3 LTS and we run our model on five different (random) seeds (6809, 36275, 5317, 82958, 25368).

The mean and standard deviation (SD) of the scores for all the competing models on different NLP tasks are reported in Table 3.3. UmlsBERT achieves the best results in four out of the five tasks. It achieves the best F1 score in three i2b2 tasks (2006, 2010, and 2012 with F1 scores 93.6%, 88.6%, and 79.4% respectively) and the best accuracy in the MedNLI task (83.0%).

As our model is initialized with the Bio_ClinicalBERT model and pre-trained on the MIMIC-III dataset, it is not surprising that it does not outperform the BERT model on the i2b2 2014 task (The $\text{BERT}_{\text{base}}$ model achieved a F1 score of 95.2% on i2b2 2014). This is probably due to the nature of the de-ID challenges [4]. Specifically, protected health information (PHI) is replaced with a sentinel 'PHI' marker in the MIMIC dataset. However, in the de-ID challenge dataset (i2b2 2014), the PHI is replaced with different synthetic masks, and thus, the sentence structure that appears in BERT's training is not present in the downstream task [4]. However, UmlsBERT achieves a better performance than the other biomedical BERT models even on this task.

These results confirm our hypothesis that augmenting contextual embedding through biomedical knowledge is beneficial for the model's performance in various biomedical downstream tasks.

Dataset		$\text{BERT}_{\text{based}}$	BioBERT	Bio_ClinicalBERT	UmlsBERT
MedNLI	Test Ac.	77.9 ± 0.6	82.2 ±0.5	81.2 ± 0.8	**83.0 ± 0.1**
	Val. Ac.	79.0 ± 0.5	83.2 ± 0.8	83.4 ± 0.9	**84.5 ± 0.1**
	R.T.(sec)	308	307	269	305
	#param.	108,312,579	108,312,579	108,312,579	108,346,371
i2b2 2006	Test F1	93.5 ± 1.4	93.3 ± 1.3	93.1 ± 1.3	**93.6 ± 0.5**
	Val. F1	94.2 ± 0.6	93.8 ± 0.3	93.4 ± 0.2	**94.4 ± 0.2**
	R.T.(sec)	12508	12807	12729	13167
	#param.	108,322,576	108,322,576	108,322,576	108,356,368
i2b2 2010	Test F1	85.2 ± 0.2	87.3 ± 0.1	87.7 ± 0.2	**88.6 ± 0.1**
	Val. F1	83.4 ± 0.3	85.2 ± 0.6	86.2 ± 0.2	**87.7 ± 0.5**
	R.T.(sec)	5325	5244	5279	5219
	#param.	108,315,655	108,315,655	108,315,655	108,349,447
i2b2 2012	Test F1	76.5 ± 0.2	77.8 ± 0.2	78.9 ± 0.1	**79.4 ± 0.1**
	Val. F1	76.2 ± 0.7	78.1 ± 0.5	77.1 ± 0.4	**78.3 ± 0.4**
	R.T.(sec)	2413	2387	2403	2432
	#param.	108,320,269	108,320,269	108,320,269	108,354,061
i2b2 2014	Test F1	**95.2 ± 0.1**	94.6 ± 0.2	94.3 ± 0.2	94.9 ± 0.1
	Val. F1	**94.5 ± 0.4**	93.9 ± 0.5	93.0 ± 0.3	94.3 ± 0.5
	R.T.(sec)	16738	17079	16643	16554
	#param.	108,343,339	108,343,339	108,343,339	108,377,131

Table 3.3: Results of mean ± standard deviation of five runs from each model on the test and the validation test; we use the abbreviation Ac. for accuracy, R. T. for running time and #param. for number of parameters; best values are **bolded**.

Effect of semantic type embeddings

In order to understand the effect that semantic type embeddings have on the model's performance, we conduct an ablation test comparing the performance of two variations of the UmlsBERT model. In one model, the semantic type embeddings are available, while in the other, they are not. The results of this comparison are listed in Table 3.4. We observe that UmlsBert achieves its best performance for every dataset when semantic type embeddings are available. This experiment further confirms the positive effect of the semantic type embeddings on the performance of the UmlsBERT model.

Dataset		UmlsBERT$_{-ST}$	UmlsBERT
MedNLI	Ac.	82.3 ± 0.2	83.0 ± 0.1
i2b2 2006	F1	93.3 ± 0.7	93.6 ± 0.5
i2b2 2010	F1	88.3 ± 0.3	88.6 ± 0.1
i2b2 2012	F1	79.1 ± 0.2	79.4 ± 0.1
i2b2 2014	F1	94.7 ± 0.1	94.9 ± 0.1

Table 3.4: Results of mean ± standard deviation of five runs for both variations of UmlsBERT on the test sets of all the datasets; In UmlsBERT$_{-ST}$, the semantic type embeddings are not available.

3.5.2 Qualitative Embedding Comparisons

Table 3.5 shows the nearest neighbors for six words from three semantic categories using UmlsBERT, Bio_ClinicalBERT, BioBERT and BERT. The first two categories ('ANATOMY' and 'DISORDER') are chosen to demonstrate the ability of the models to identify similar words in a clinical context and the third category ('GENERIC') is used to validate that the medical-focus BERT models are able to find meaningful associations between words in a general domain, even if they are trained on medical-domain text datasets.

	ANATOMY		**DISORDER**		**GENERIC**	
	feet	kidney	mass	bleeding	school	war
BERT$_{\text{based}}$	ft	liver	masses	bleed	college	battle
	foot	lung	massive	sweating	university	conflict
BioBERT	foot	liver	masses	bleed	college	wartime
	wrists	lung	weight	strokes	schooling	battle
Bio_ClinicalBERT	foot	liver	masses	bleed	college	warfare
	legs	lung	weight	bloody	university	wartime
UmlsBERT	foot	**Ren**	**lump**	bleed	college	warfare
	pedal	liver	masses	**hem**	students	military

Table 3.5: The two nearest neighbors for six words in three semantic categories (two clinical and one generic). Note that only UmlsBERT finds word associations based on the CUIs of the UMLS Metathesaurus that have clinical meaning, whereas in the generic category there are no discernible discrepancies between the models.

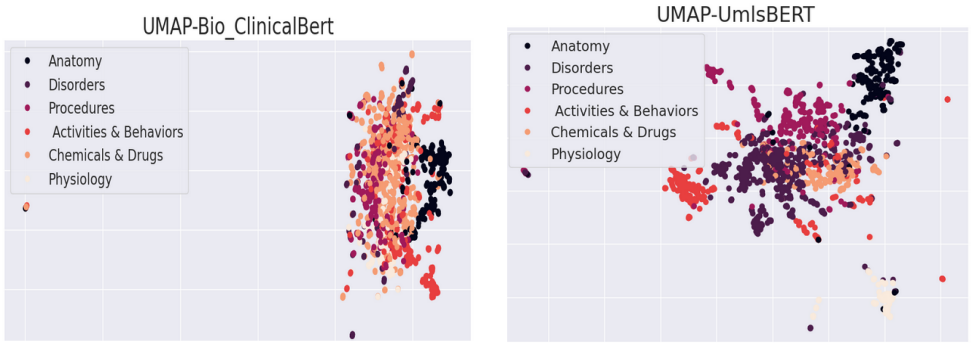
This analysis demonstrates that augmenting the contextual embedding of UmlsBERT

with medical information (from the UMLS Metathesaurus) is indeed beneficial for discovering associations between words with similar meanings in a clinical context. For instance, only UmlsBERT discoverers the connection between 'kidney' and 'ren' (from the Latin word 'renes', which means kidneys), between 'mass' and 'lump', between 'bleeding' and 'hem' (a commonly used prefix to refer to blood) and between 'feet' and 'pedal' (an adjective meaning 'pertaining to the foot or feet' in a medical context).

These associations result from changing the nature of the Masked LM training phase of UmlsBERT to a multi-label scenario by connecting different words that share a common CUI. In the previously mentioned examples, 'kidney' and 'ren' have *CUI:C0022646*; 'mass' and 'lump' have *CUI:C0577559*; 'bleeding' and 'hem' have *CUI:C0019080*; and 'feet' and 'pedal' have *CUI:C0016504*.

The generic list of words indicates that the medical-focused BERT models do not trade off their ability to find meaningful associations in a general domain for more precision in a clinical context. This is based on the fact that there is no meaningful difference observed in the list of neighbor words that the four models identified.

3.5.3 Semantic Type Embedding Visualization



(a) (b)

Figure 3.4: UMAP visualization of the clustering **(a)** of the Bio_ClinicalBert input embedding (word embedding) **(b)** of the UmlsBert input embedding (word embedding + semantic type embedding).

In order to demonstrate the effect of the semantic types on the creation of the model's input word embeddings, Figure 3.4 presents a UMAP dimensionality reduction [68] mapping

comparison between Bio_ClinicalBERT and UmlsBERT. We compare the input embedding (word embedding) of Bio_ClinicalBERT with the input embedding (word embedding + semantic type embedding) of UmlsBERT for all the clinical terms that UMLS identified in the standard BERT vocabulary. In the graph, we group the medical terms by their semantic groups, which are more general clusters consisting of different semantic types. For example, the semantic types 'Cell' and 'Body System' belong in the semantic group 'ANATOMY' (a more detailed description of the semantic groups and the semantic types is provided in appendix B).

Evidently, the clustering according to the semantic group that exists in the UmlsBERT embeddings (Figure 3.4b) cannot be found in the Bio_ClinicalBERT embeddings (Figure 3.4a). Thus, we can conclude that more meaningful input embeddings can be provided to the model by augmenting the input layer of the BERT architecture with the semantic type vectors. This is because they can force the embeddings of the words of the same semantic type to become more similar to each other in the embedding space.

3.6 Conclusion

In this chapter, we presented a novel BERT-based architecture that could incorporate domain (biomedical) knowledge in its pre-training process and had the ability to learn more easily the associations between distinctive terminologies. In particular, we enhanced the input layer of a contextual model with semantic type knowledge of the medical words in the input sentence. We also updated the Masked Language Modelling pre-trained task to take into consideration the connection between medical words that have the same underlying 'concept' in UMLS.

Our experiments, conducted in different clinical named entity recognition (NER) tasks as well as in one clinical natural language inference task, indicated that the features described above had a positive effect on the performance of a medical contextual model. Our qualitative analysis also established that the model could learn more easily the association of different clinical terms with similar meaning in the UMLS Metathesaurus. Finally, by leveraging information from the semantic types of each (biomedical) word, our model could create more meaningful input embeddings, as it forced the embeddings of the words of the same semantic type to become more similar to each other in the embedding space.

In the following chapter, we will investigate whether extending the medical features to a sequence-to-sequence contextual model can positively affect the model's performance in summarizing medical conversations by guiding the summarization process to include relevant medical facts in the summarized output.

Chapter 4

Augmenting Transformer-based Sequence-to-Sequence Model for Summarizing Medical Conversations

4.1 Introduction

In Chapter 3, we analyzed how the architecture of a contextual encoder model can be augmented with structured knowledge from a medical database. We demonstrated that this information could aid the model in learning the associations between distinctive terminologies.

In this chapter, we will extend these medical features and demonstrate how medical structured knowledge that is integrated into a summarization model can guide the summarization process to include relevant medical facts in the summarized output. We will also investigate whether integrating medical guidance signals into a summarization architecture can boost the performance of a transformer-based sequence-to-sequence model in the task of summarizing medical conversations.

In particular, we will explore the effect of a novel medical guidance signal, which consists of all the medical words of the input sentence, on the model's performance. We will also illustrate that augmenting the input layer of a sequence-to-sequence model with medical knowledge can be beneficial for the model's performance. This is akin to how we showed the positive effect of augmenting the input layer of an encoder transformer model with medical information in Chapter 3. We will also provide a new loss function that provides the model a stronger incentive to predict medical words.

We will demonstrate that these features can facilitate the model to achieve state-of-the-art results on multiple medical conversation summarization datasets. Also, our qualitative analysis will validate that our model can provide summaries containing relevant medical facts and thus can help with the omission of key information problem. This is especially of concern in the medical domain because if key medical information is missing from the output, future readers may be unable to make an accurate diagnosis.

The remainder of this chapter is organized into five parts. Section 4.2 presents related work, followed by details of the characteristics of the proposed architecture (MedicalSum) for integrating clinical knowledge into a summarization model in Section 4.3. The experimental setup and data that are used to train and test the MedicalSum model are described in Section 4.4. The results of the experiments and the qualitative analysis are reported in Section 4.5, followed by the conclusion of this chapter in Section 4.6.

4.2 Related Work

There are two main approaches for summarization, namely: (i) extractive methods, where the summary is created from passages that are copied from the source text [55]; and (ii) abstractive methods, where phrases and words not in the source text can be used to create the summary [21].

Neural Abstractive Summarization: For the task of abstractive summarization, sequence-to-sequence (seq-to-seq) summarization models have achieved state-of-the-art results [101]. As mentioned in Chapter 1, the sequence-to-sequence models were a special class of Recurrent Neural Network architectures that map the input sequence to the output sequence [101]. These models were composed of an encoder and a decoder. The task of an encoder network was to understand the input sequence and then generated a compact representation. With such representation at hand, the decoder could then generate a target sequence.

Different architectures have been proposed to improve the performance of a seq-to-seq model. See et al. [94] used a pointing mechanism for copying words from the source document (Figure 4.1). Enarvi et al. [27] incorporated a transformer-based [113] encoder-decoder architecture with a pointing mechanism in order to produce highly-accurate summaries. However, the main shortcoming of these approaches was that they did not take advantage of any external structured information to produce more accurate summaries.

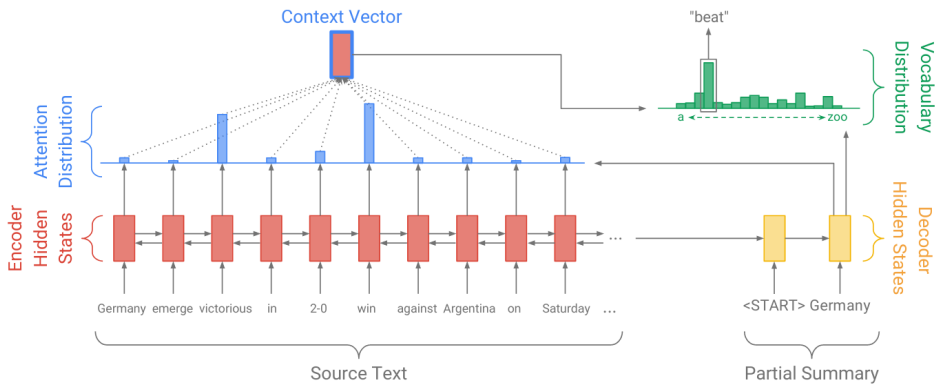


Figure 4.1: The pointing architecture for copying words of See et al. [94].

Guided Summarization: Several studies have focused on including guidance signals in the standard seq-to-seq architecture. Li et al. [60] included a set of keywords that were incorporated into the generation process. Zhu et al. [128] proposed the usage of relational triples (subject, relation, and object). Dou et al. [26] created a guided summarization framework that can support different external guidance signals (e.g., keywords, highlighted sentences, and relations). However, the models mentioned above did not take advantage of structured medical information that exists in external databases.

Medical Summarization: Pivovarov et al. [84] introduced a summarization model which was focused on creating accurate summaries for clinical data, and Zhang et al. [124] employed a model with a pointing mechanism to generate summaries from radiology reports. Enarvi et al. [27] utilized a pointer-generator transformer model to accurately generate notes from doctor-patient conversations. Joshi et al. [49] relied a variation of the pointer-generator model that leveraged shared medical terminology between source and target to distinguish important words from unimportant ones.

However, these models have not taken advantage of structured medical information in their decision process, which could help key information pass the model's decision process and appear in the summary. This is especially of concern in the medical domain, as inaccuracies could significantly affect future patient health outcomes.

4.3 Method

To enhance the performance of previous approaches we developed MedicalSum, which we demonstrate can successfully integrate medical knowledge with a sequence-to-sequence contextual model to guide the summarization process in various ways with a view to increasing the likelihood of relevant medical facts being included in the summarized output.

4.3.1 MedicalSum: Medical Guided Transformer Pointer Generator Model

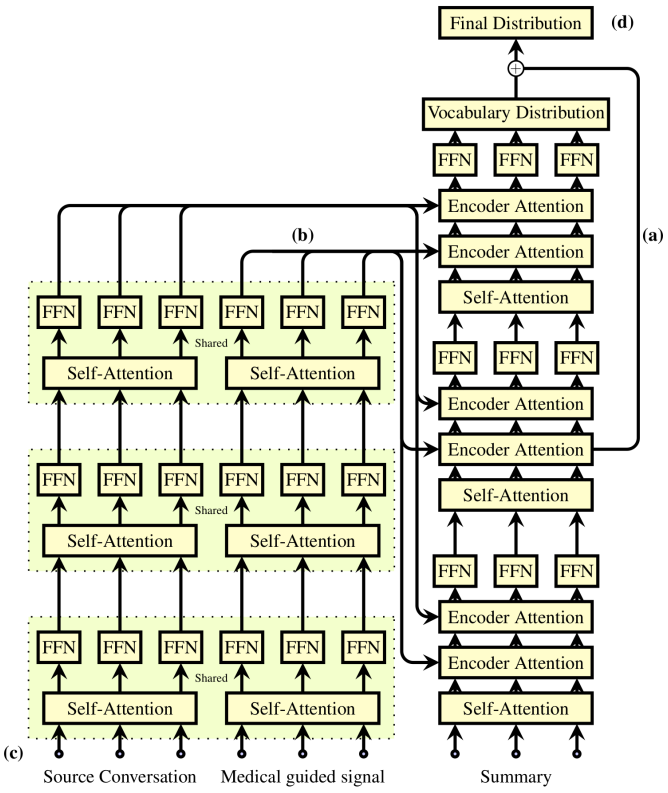


Figure 4.2: Illustration of MedicalSum a transformer sequence-to-sequence model with a pointer-generator and guidance mechanism. MedicalSum also introduces a new semantic type embedding and a novel weighted loss function.

A simplified image of the MedicalSum model can be found in Figure 4.2. We adopt the transformer self-attention model [113] in both the encoder and decoder to create context-dependent representations of the inputs. Both the encoder and the decoder consist of six self-attention layers with eight attention heads. Each decoder layer attends to the top of the encoder stack after the self-attention. Each encoder and decoder layer contains a feed-forward layer with a ReLU activation between two transformations. Following Enarvi et al. [27], we apply layer normalization [9] before the feed-forward and the self-attention sub-layers.

We improve the performance of our model by introducing the following additions into the standard transformer encoder-decoder model for summarization:

1. a pointing mechanism for copying out-of-vocabulary (OOV) words from the source document (part (a) in Figure 4.2);
2. a novel guided summarization signal which consists of all the medical words in the input sentence in UMLS (part (b) in Figure 4.2);
3. a new semantic type embedding that enriches the input embeddings process (part (c) in Figure 4.2); and
4. a novel weighted loss function which provides the model a stronger incentive to correctly predict medical words (part (d) in Figure 4.2).

The details of each added component are discussed in the following sections.

4.3.2 Pointer-Generator

We implement the pointer generator network as described by both Enarvi et al. [27] and See et al. [94]. Because the transformer model creates several encoder-decoder attention distributions, we can choose any distribution over the source tokens for the copying mechanism. Following Enarvi et al. [27], we choose to train only the parameters of a single head to attend to the tokens that are good candidates for copying. In contrast, the rest of the attention heads are left to perform their usual function. In Garg et al. [30], it was stated that the penultimate layer seems to learn alignments naturally, so we decide to use its first attention head for pointing [27].

4.3.3 Medical Guidance Signal

We include a medical guidance signal in the summarization process which comprises all the medical terms in the input sequence that could be identified in UMLS using the MedCAT toolkit [52]. The inclusion of the medical signal is accomplished by introducing two encoders (that share weights) to encode the input text and the guidance signal, respectively [26]. Each encoder layer, for both the input and the guidance signal, consists of a self-attention block and a feed-forward block (equation 4.1).

$$\begin{aligned} x &= LN(x + SelfAtnn(x)) \\ x &= LN(x + FeedForward(x)) \end{aligned} \tag{4.1}$$

Each decoder layer consists of: (i) a self-attention block; (ii) a cross-attention block with the medical guidance signal (mg) to inform the decoder which sections of the source document are important; and (iii) a cross-attention block with the encoded input (x_{in}), where the decoder attends to the whole source document based on the guidance-aware representations and a feed-forward block (equation 4.2).

$$\begin{aligned} y &= LN(y + SelfAtnn(y)) \\ y &= LN(y + CrossAtnn(y, mg)) \\ y &= LN(y + CrossAtnn(y, x_{in})) \\ y &= LN(y + FeedForward(y)) \end{aligned} \tag{4.2}$$

As MedicalSum focuses on medical data summarization, we generate a medical guidance signal with all the words with a medical meaning. We believe that this signal will be beneficial for the performance of the model since a guidance signal which is created as a set of individual keywords $\{w_1, ..., w_n\}$ can help the model to focus on specific desired aspects of the input [26]. As mentioned in Chapter 3, we choose to identify medical entities defined in the UMLS Metathesaurus, which is a compendium of many biomedical (e.g. MeSH [24], ICD-10 [76]) and thus includes all the major standardized clinical terminologies. In order to exclude less relevant entities, we follow the strategy proposed by Adams et al. [1] and we only keep the entities from specific semantic types in the UMLS 2020AA version that are relevant to the domain of the dataset (i.e., Anatomy, Disorders, Chemicals & Drugs, and Procedures).

4.3.4 Semantic Type Embeddings

We also introduce a new embedding matrix called $S \in \mathbb{R}^{D_s \times d}$ into the input embedding layer where d is the transformer hidden dimension and $D_s = 50$ is the number of unique UMLS semantic types that are relevant to the domain of the dataset. In the S matrix, each row represents the unique semantic type in UMLS that a word can be identified with.

To incorporate the S embedding matrix into the input embedding layer, all the words with a clinical meaning defined in UMLS are identified (using the MedCAT toolkit [52]), and their corresponding semantic type is extracted. By introducing the semantic type embedding, the input vector for each word w_j is updated to (equation 4.3):

$$u_{input}^{(j)\prime} = p^{(j)} + Ew_j + S^\top s_{wj} \tag{4.3}$$

where $s_{wj} \in \mathbb{R}^{D_s}$ is a 1-hot vector corresponding to the semantic type of the medical word w_j (the semantic type vector $S^\top s_{wj}$ is set to a zero-filled vector for words that are not identified in UMLS) and $p^{(j)} \in \mathbb{R}^d$ is the position embedding of the j^{th} token in the sentence. Finally, $E \in \mathbb{R}^{d \times D}$ is the token embedding, where D is the size of the model's vocabulary and $w_j \in \mathbb{R}^D$ is a 1-hot vector corresponding to the j^{th} input token.

In Chapter 3, we demonstrated that the inclusion of semantic type vectors could enhance on the performance of an encoder transformer-based model in various downstream tasks. The semantic type embeddings could provide more accurate input vectors for the medical words that are rare in the training corpus and the model may not have the chance to learn meaningful information for their representations.

In this chapter, we will investigate whether including semantic types information into the input layer can enrich the input embeddings of a sequence-to-sequence model by forcing the embeddings of words associated with the same semantic type to become closer to each other in the embedding space. It should be noted that in our experiments, we enrich the input embedding of both the input encoder and the guidance encoder.

4.3.5 Medical Weighted Loss Function

We update the loss function of the summarization task in order to provide a stronger incentive to predict words with a medical meaning correctly. In our summarization model, we use the cross-entropy loss of the Fairseq library [77] for the target word x_t for each timestep t. We modify the loss function to a weighted loss function where the weight for

all medical words is higher. Specifically, the summarization loss is updated to (equation 4.4):

$$loss = -logP(x_t) * w_t \tag{4.4}$$

where $w_t = 1$ for all the non-medical words and $w_t = 1 + \alpha$ for all the words with a medical meaning, in which α is an additional weight value for these words.

4.4 Experiments

This section presents the results of an empirical evaluation of the MedicalSum model. To demonstrate the efficiency of our proposed model, we will provide a comparison between the MedicalSum and the pointer generator transformer model of Enarvi et al. [27] (which has achieved state-of-the-art results in medical summarization). We will also present the results of an ablation test to examine the effect of each medical signal on the performance of the model. We will also provide a qualitative output comparison to illustrate how the inclusion of medical knowledge can improve the quality of medical summaries.

4.4.1 Dataset

For the training of the MedicalSum model, we have to select a large enough dataset that would provide the necessary data for the medical signals to meaningfully affect the model's performance. However, there are no publicly available large-scale datasets for medical summarization, and thus, we use a proprietary one. We use English language data consisting of encounters in a family medicine setting. The data are recorded at the time of the encounter and they also include associated clinical note summaries. It should be noted that the conversation transcripts of the audio files are obtained using an automatic speech recognizer [27].

The reports are organized under three sections corresponding to three broad areas of a medical note as follows:

1. History of Present Illness (HPI), which captures the reason for the visit, and the relevant clinical and social history.

2. Physical Examination (PE), which captures both normal and abnormal findings from a physical examination.

3. Assessment and Plan (AP), which captures the assessment by the doctor and the treatment plan, e.g., medications and physical therapy.

We evaluate our summarization models on the creation of the summaries for each section. The experimental results are based on a dataset that consists of around 40,000 encounters for each section. The dataset is partitioned chronologically (date of collection) into training, validation, and testing partitions. It should be noted that the doctors present in the testing set are also present in the training set. Table 4.1 shows detailed statistics of our dataset in terms of the number of training examples and source and target sequence lengths.

	Train	Valid	Test	A.W	P.D (%)
AP	42106	648	2525	2586	99.2
HPI	43092	657	2551	2584	96.9
PE	39815	635	2442	2633	91.7
RAD	91544	2000	600	49	100

Table 4.1: Number of reports/encounters for the train/validation/test set of each section of the family medicine reports and the MEDIQA third task; P.D is the percent distribution of encounters which have the section in their report, and A.W is average word count in those encounters.

As previously mentioned, there are no large-scale public datasets for medical conversation summarization. For a more open comparison, we also experiment with a public dataset. We tackle the third task of the MEDIQA 2021 challenge [16] of automatic summarization of English radiology reports (RAD) of the MIMIC-CXR dataset [47] (license: https://tinyurl.com/mimic-licence). From Table 4.1, it can be observed that the input documents in the MEDIQA dataset are much smaller than the documents of the other real-world datasets on which we experiment and thus contain less medical information. However, we include this dataset in order to have an evaluation of the models and the baseline on a publicly available dataset.

4.4.2 Experimental Setup

We report the results of the comparison of our proposed MedicalSum model with the baseline pointer-generator model (Enarvi-PG) [27]. We also experiment with three variations of our model that only contain (a) the guidance signal (MedicalSum$_{guidance}$); (b) the semantic type embedding (MedicalSum$_{semantic}$); and (c) the medical weighted loss function

(MedicalSum$_{loss}$), in order to measure how each signal individually affects the model's performance. These models are implemented using the Fairseq library [77] on PyTorch 1.5.0. All experiments are executed on V100 GPU with 32G GB of system RAM on Ubuntu 18.04.3 LTS.

We use a vocabulary consisting of the 45k most frequent words. The same vocabulary is shared between the source and the target tokens. We train the models for a maximum of 20k steps. It should be noted that Enarvi-PG, the MedicalSum$_{guidance}$ and MedicalSum$_{loss}$ model have the exact same number of parameters (74,724,353), as the input and the 'guidance' encoder share their weights. However, MedicalSum and the MedicalSum$_{semantic}$ model have an additional 25,600 parameters due to the inclusion of the semantic type embeddings.

Hyperparameter tuning

In order to address the reproducibility concerns of the NLP community [25], we provide the search strategy and the bound for each hyperparameter: the batch size is set between 4 and 8, and the α parameter of the medical weight loss is tested with the values 0.01, 0.1 and 0.2. The best values are chosen based on the validation set micro ROUGE-1 F1 values. To make a fair comparison, we tune the hyperparameters of each model in order to demonstrate its best performance. For the Enarvi-PG, MedicalSum, and the models with each individual medical signal, the batch size is set to 4, and the medical weight loss parameter is set to 0.01.

We run our models on three (random) seeds, and we provide the average scores and standard deviation for the testing and the validation set. We compare the models on the ROUGE-1 F1 score, which is based on the overlap of unigram, and the ROUGE-L F1 score, which is based on the lengths of the longest common subsequences between the actual summary and the output of the model.

4.5 Results

4.5.1 Summarization Model Comparison

The mean and standard deviation of ROUGE-1 F1 and ROUGE-L F1 for all the competing models are reported in Table 4.2.

MedicalSum outperforms the pointer generator (Enarvi-PG) baseline on all the datasets since all the (three) previously mentioned medical signals have a positive contribution to

TEST					
Model	Micro F1	HPI	PE	AP	RAD
Enarvi-PG	Rouge-1	48.04 ± 0.4	66.11 ± 0.3	43.02 ± 0.4	27.01 ± 0.2
	Rouge-L	34.21 ± 0.3	63.15 ± 0.2	36.19 ± 0.3	25.01 ± 0.3
$Med.Sum_{loss}$	Rouge-1	48.64 ± 0.2	67.37 ± 0.2	43.85 ± 0.4	27.34 ± 0.2
	Rouge-L	34.32 ± 0.3	63.77 ± 0.3	36.67 ± 0.5	25.37 ± 0.2
$Med.Sum_{guid.}$	Rouge-1	48.79 ± 0.3	68.02 ± 0.2	43.72 ± 0.5	27.57 ± 0.2
	Rouge-L	35.14 ± 0.3	64.17 ± 0.2	36.65 ± 0.3	25.66 ± 0.2
$Med.Sum_{sem.}$	Rouge-1	48.90 ± 0.2	67.80 ± 0.3	43.64 ± 0.4	27.56 ± 0.3
	Rouge-L	34.79 ± 0.2	63.93 ± 0.2	36.42 ± 0.2	25.39 ± 0.3
MedicalSum	Rouge-1	**48.98 ± 0.3**	**68.22 ± 0.2**	**44.54 ± 0.3**	**27.77 ± 0.3**
	Rouge-L	**35.22 ± 0.3**	**64.48 ± 0.3**	**37.34 ± 0.2**	**26.06 ± 0.2**
VALIDATION					
Enarvi-PG	Rouge-1	48.17 ± 0.3	67.44 ± 0.2	43.23 ± 0.4	29.91 ± 0.3
	Rouge-L	34.88 ± 0.3	64.68 ± 0.2	36.39 ± 0.3	29.95 ± 0.3
$Med.Sum_{loss}$	Rouge-1	49.29 ± 0.2	67.89 ± 0.2	44.02 ± 0.3	30.32 ± 0.3
	Rouge-L	34.94 ± 0.3	64.33 ± 0.3	36.70 ± 0.2	30.14 ± 0.3
$Med.Sum_{guid.}$	Rouge-1	49.55 ± 0.3	68.18 ± 0.3	44.32 ± 0.4	30.35 ± 0.2
	Rouge-L	35.14 ± 0.3	64.66 ± 0.2	37.01 ± 0.3	30.81 ± 0.2
$Med.Sum_{sem.}$	Rouge-1	49.39 ± 0.3	68.02 ± 0.2	44.16 ± 0.4	30.30 ± 0.2
	Rouge-L	34.99 ± 0.4	64.41 ± 0.3	36.90 ± 0.5	30.50 ± 0.2
MedicalSum	Rouge-1	**49.68 ± 0.2**	**68.37 ± 0.3**	**44.98 ± 0.3**	**30.63 ± 0.3**
	Rouge-L	**35.43 ± 0.2**	**64.83 ± 0.2**	**37.90 ± 0.2**	**31.45 ± 0.3**

Table 4.2: Results of mean ± standard deviation for each model on the test/validation set; best values are **bolded**.

its performance (see ablation details in subsection 4.5.2) by encouraging MedicalSum to take into consideration different medical information (subsection 4.5.3). It achieves an improvement of between 0.8% (on the radiology dataset) and 2% (on the PE section). The $\text{MedicalSum}_{semantic}$, the MedicalSum_{loss}, and the Enarvi-PG model have similar running times (117K seconds for the HPI, AP and PE sections and 64K seconds for the radiology dataset). MedicalSum and the $\text{MedicalSum}_{guidance}$ are always slower (by 4%) due to the introduction of the second 'guidance' encoder.

4.5.2 Ablation Study

In order to understand the effect that each medical signal has on the model's performance, we conduct an ablation test, comparing the performance of three variations of the MedicalSum model, with each model being allowed to access only one of the medical signals. The results of this comparison are listed in Table 4.2. We observe that for every dataset, MedicalSum achieves its best performance when all the medical signals are available. However, as can be observed in Table 4.2, each model that has access to any of the medical signals outperforms the baseline model.

The guidance signal (MedicalSum$_{guidance}$) seems to have the most positive effect across all the sections and the radiology dataset since it can more clearly guide the model to the most important sections of the input. On the other hand, the medical weight loss (MedicalSum$_{loss}$) seems to have the least influence on the model's performance. However, as we will show in the qualitative analysis (subsection 4.5.3), it can aid the model in focusing on medical information. Enriching the input embedding with semantic information (MedicalSum$_{semantic}$) seems to boost the performance of the model as it forces the embeddings of words that are associated with the same semantic type to become more similar to each other in the embedding space. These experiments provide further evidence of the positive effect of each of the medical signals on the performance of the MedicalSum model. Finally, in subsection 4.5.3, we will provide a qualitative analysis to demonstrate how each signal can 'guide' a summarization model to medical information that the baseline model overlooked.

4.5.3 Qualitative Model Output Comparison

In the previous section, we showed how the MedicalSum model improves ROUGE scores compared to a state-of-the-art baseline. In this section, we will qualitatively evaluate some of the differences in summaries produced by each model variant and illustrate how each feature contributes to the creation of accurate medical summaries.

To do this, we sample several generated outputs and illustrate the differences between the different medical models. Table 4.3 shows a sample target summary and the comparison with the output of the Enarvi-PG transformer model and the medical signals of MedicalSum. It should be noted that the only criterion for choosing these examples is that they should contain medical entities. Some of the shorter input texts contain no medical entities. Thus, we do not include those texts here as they would not show the differences between the baseline and the guided medical summarization model. As it can be observed

Model	Output
Target-summary	ASSESSMENT AND PLAN –lf1– Acute left hip pain . –lf1– The patient has been experiencing acute left hip pain since . The pain is in her sacroiliac joint on her left side . I have prescribed a short course of prednisone to take in the morning after breakfast . I have instructed her not to take **ibuprofen, Advil , or Aleve** . I have also instructed her to apply heat to her back . we can get an **x-ray** of her lower back.
Enarvi-PG	ASSESSMENT AND PLAN –lf2– 1 . Left hip pain . –lf1– . I recommend the patient take a short course of prednisone . I have advised her to apply heat to the affected area . –lf2– 2 . Preventative care .–lf2– Instructions –lf1–
$Medical\text{-}Sum_{semantic}$	Assessment and Plan : –lf2– 1 . Acute left hip pain . –lf1– the left hip pain is sciatic pain. I recommend the patient take a short course of prednisone for 10 days . She can take it in the morning after breakfast . I also recommend she take a shower going to the kitchen daily . I also recommend she can also try applying heat to the affected area .
$Medical\text{-}Sum_{guidance}$	ASSESSMENT AND PLAN –lf2– 1 . Acute left hip pain . –lf1– I have prescribed a short course of prednisone . I advised the patient to take this in the morning after breakfast . I advised the patient to apply heat to the affected area for 10 days .advised the patient to avoid **NSAIDs** . I have ordered an **x-ray** of the lower back .
$Medical\text{-}Sum_{loss}$	ASSESSMENT AND PLAN –lf2– 1 . Acute left hip pain . –lf1– I have prescribed her a short course of prednisone . If her symptoms do not improve , I will refer her to physical therapy to strengthen the muscle of her back . –lf2– 2 . Preventative care . –lf1– The patient is due for a **x-ray** .
$MedicalSum$	ASSESSMENT AND PLAN –lf1– 1 . Acute left hip pain . –lf1– I suspect the left hip pain is sciatic pain . I will order an **x-ray** of the left hip . I have instructed the patient to apply heat to the lower back and see unusual daily chores . I advised the patient to avoid **NSAIDs** I recommended that she perform acute exercise when she is acutely inflamed after 10 days.

Table 4.3: First example of distinct output from summarization models of different medical signals. The medical entities are highlighted in red color.

in Table 4.3, the summary from the pointer generator (Enarvi-PG) baseline model does

not include the following medical information:

1. the medicines: ibuprofen, Advil, or Aleve and
2. x-rays

. In contrast, the MedicalSum$_{guidance}$ model has included in its summary the medical entities x-ray and NSAID (Non-steroidal anti-inflammatory drugs), which are identified as medical entities and were part of the guidance signal. The MedicalSum$_{loss}$ model also includes the x-ray token since it provides a stronger incentive for the model to predict 'medical' words by assigning a higher weight to the words that have a medical meaning. The MedicalSum model includes all the previously mentioned medical words, as it has taken advantage of all the medical signals.

Model	Output
Target-summary	The left eye swelling with burning and itching has been present for 2 weeks . I will prescribe olopatadine 0.2 percent **ophthalmic drops** . If the symptoms do not improve , the patient will return and consider bacterial cause .–lf2– **Dyslipidemia** .
Enarvi-PG	ASSESSMENT AND PLAN : The patient has allergic conjunctivitis of left eye . –lf1– - Patient Education and Counseling : The patient was advised to keep wiping green and thick mucus from the eye
MedicalSum$_{semantic}$	ASSESSMENT AND PLAN : –f1– 1 . Allergic conjunctivitis of left eye. –f1– The patient was advised to continue using his **eye drops** .
MedicalSum$_{guidance}$	ASSESSMENT/PLAN –f1– Allergic conjunctivitis . I recommended that the patient continue daily loratadine .–lf2– **Dyslipidemia**
MedicalSum$_{loss}$	ASSESSMENT/PLAN –lf1– Allergic conjunctivitis –lf1– –lf1– Medical Treatment : The patient will continue to take loratadine –lf2– **Dyslipidemia**
MedicalSum	ASSESSMENT/PLAN –f1– Allergic conjunctivitis . –f1– - Medical Treatment : The patient will continue daily loratadine and **eye drops** .–lf2– **Dyslipidemia**

Table 4.4: Second example of distinct output from summarization models of different medical signals. The medical entities are highlighted in red color.

Another example that showcases the strength of the medical signals can be found in Table 4.4. The summary from the Enarvi-PG model lacks two important medical facts:

1. ophthalmic drops; and

2. Dyslipidemia.

Both the MedicalSum$_{guidance}$ and the MedicalSum$_{loss}$ models include the word 'Dyslipidemia', as it is identified as a medical word. The MedicalSum$_{semantic}$ model also includes the medical concept 'eye drops' as a replacement for 'ophthalmic drops' since 'eye' and 'ophthalmic' have the same semantic type in UMLS. The MedicalSum model includes all of the previously mentioned medical words. These examples demonstrate how, in addition to improving ROUGE scores, the MedicalSum model also generates clinical summaries that contain more relevant medical facts.

4.6 Conclusion

In this chapter we presented a novel approach for medical conversation summarization that integrated medical knowledge into the summarization process of a contextual model. In particular, our model could provide external medical guidance that helps key information pass the model's decision process and appear in the summary. We also introduced a novel weighted loss function that provides a stronger incentive for the model to correctly predict words with a medical meaning. The model also created more meaningful input embeddings by forcing the embeddings of the words associated with the same semantic type to become more similar to each other by incorporating information from the semantic type of each medical word into the input embedding layer of the model.

Our analysis showed that these features allowed the model to produce more accurate AI-generated medical documentation. MedicalSum outperformed the pointer-generator (Enarvi-PG) baseline and achieves ROUGE score gains of 0.8 to 2 points. Our ablation study demonstrated the positive effect of each of the medical signals on the performance of the MedicalSum model, as the model achieved its best performance for every dataset when all the medical signals were available. The qualitative analysis also showed that our model did a more complete job by including medical entities that contain crucial medical information in the output summary.

Thus far, we have shown novel techniques for augmenting the architecture of a contextual model with external structured information. In the following chapter, we will examine the effectiveness of a model that can integrate information from the relations of different labels with a contextual model that can process large documents. We will focus the research on a challenging clinical NLP multi-label classification task, namely, the ICD automatic coding problem.

Chapter 5

Knowledge Augmentation of Contextual Models for Imbalanced Multi-Label Classification Problems in the Biomedical Domain

5.1 Introduction

In the previous chapters, we proposed several novel techniques for augmenting the standard transformer-based architecture with structured information from knowledge bases. We showed how external general lexical knowledge could aid a model in distinguishing which words are semantically similar and how structured medical knowledge can boost the performance of transformer-based models in a variety of medical tasks. In this chapter, we will investigate whether integrating a contextual model with information from the relations of different labels with a novel attention mechanism can boost its performance in a multi-label classification problem.

We will investigate the case of the International Classification of Diseases (ICD) coding classification task. The ICD system is a widely used coding system, maintained by the World Health Organization [8]. It contains a special set of alphanumeric codes [50] representing diagnosis and treatment procedures during a patient visit to a healthcare facility. The ICD system is mainly used for billing and reporting purposes [43]. However, it can also be used to codify related information such as symptoms, cause of injury and patient complaints.

ICD codes have improved the consistency across physicians in recording patient symptoms and diagnoses for the purposes of clinical research and payer claims. Assigning the most appropriate codes is an important task in healthcare since erroneous ICD codes could seriously affect the organization's ability to measure the patient outcome accurately[46].

The ICD coding system organizes codes in a tree structure, with edges representing is-a relationships between parents and children from the most general to the most specific codes that are accompanied with non-essential modifiers (Figure 5.1). By treating the medical coding of medical documents as a multi-label text classification task (i.e., assigning a set of labels to each instance) we will investigate whether the relations between different labels can be efficiently encoded in an attention mechanism. Specifically, we will consider whether integrating information from a Graph Convolutional Network (GCN) which takes advantage of the relations between different medical codes, with a BigBird contextual model that can process large documents can boost the performance of a model on the medical coding classification task.

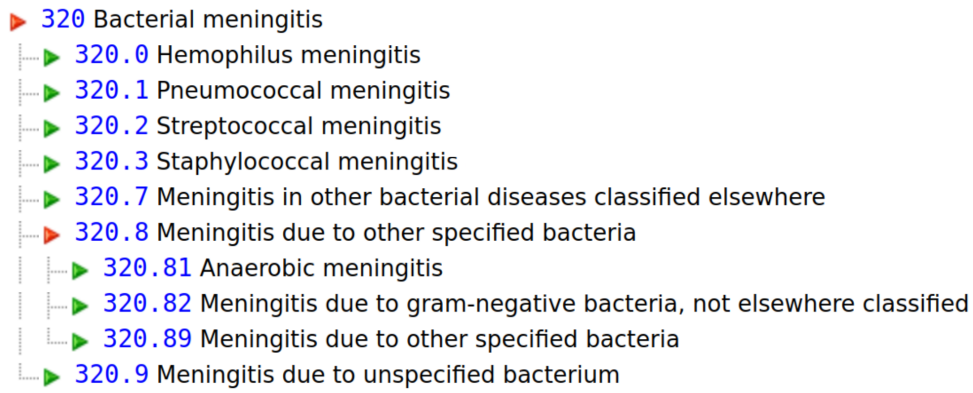


Figure 5.1: Example of the hierarchical nature of the ICD codes [105].

This chapter is organized into five parts. Related work on ICD coding is presented in Section 5.2. The characteristics of the proposed architecture for integrating clinical knowledge into a contextual text classification model are detailed in Section 5.3. The dataset and the experimental setup are described in Section 5.4. The results of the experiments and our ablation study are reported in Section 5.5, followed by the conclusion of this chapter in Section 5.6.

5.2 Related Work

The ICD coding task is a crucial task for making accurate clinical, operational, and financial decisions in healthcare. Traditionally, medical coders review clinical documents and manually assign the appropriate ICD codes by following specific coding guidelines.

Recent development in NLP has introduced deep learning models that achieved state-of-the-art performance on the ICD classification task. Shi et al. [96] proposed a model that used word/character embeddings and recurrent neural networks (LSTM) to generate the representations of the ICD codes. Mullenbach et al. [74] introduced the CAML model, which used an attention-based convolutional neural network (CNN) model with an attention mechanism in order to identify the most relevant segments to the ICD codes in each medical note (Figure 5.2). However, the main disadvantage of these approaches was that they did not consider the relations between the different codes.

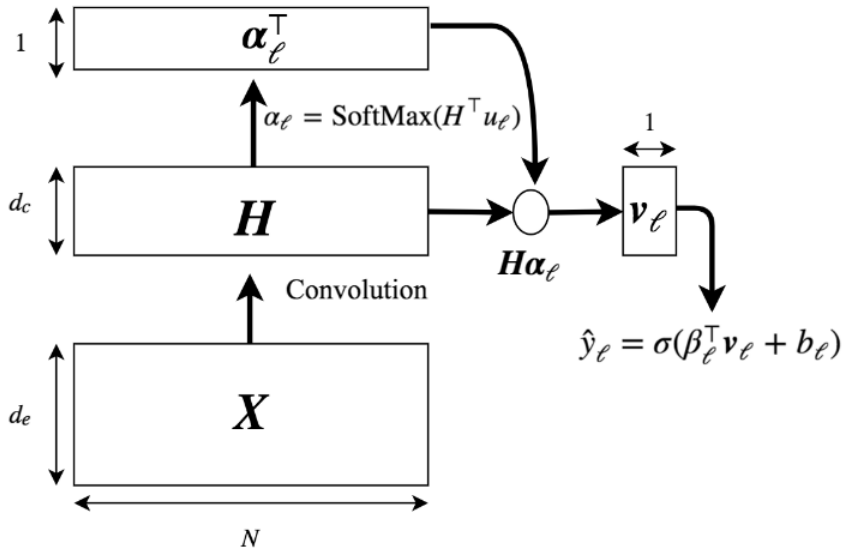


Figure 5.2: The CAML architecture in [74].

Cao et al. [20] used the co-appearance values between the ICD codes for creating a weighted adjacency matrix in order to exploit the co-appearance between the codes in the medical notes. However, the main disadvantage of using this metric was that it cannot accurately calculate the relation between two highly correlated but 'unpopular' codes.

Recent attempts at using contextual models (e.g., BERT [23]) on the ICD classification task have failed to achieve state-of-the-art results [125] mainly due to their inability to process long documents (i.e., medical notes). Fortunately, Zaheer et al. [122] introduced the BigBird model, a contextual model that allows the processing of large documents. The main advantage of this model is the use of a combination of three types of sparse attention mechanisms as illustrated in Figure 5.3, namely, (a) Random attention since two tokens that are in different positions may still share useful information; (b) Window attention where a full spectrum of attention to n-nearest tokens for each token is guaranteed; and (c) Global attention since global tokens can be used to represent the entire input sequence. This combined sparse attention mechanism allows the model to address one of the main limitations of contextual models like BERT [23], which is the quadratic dependency on the sequence length due to their full attention mechanism.

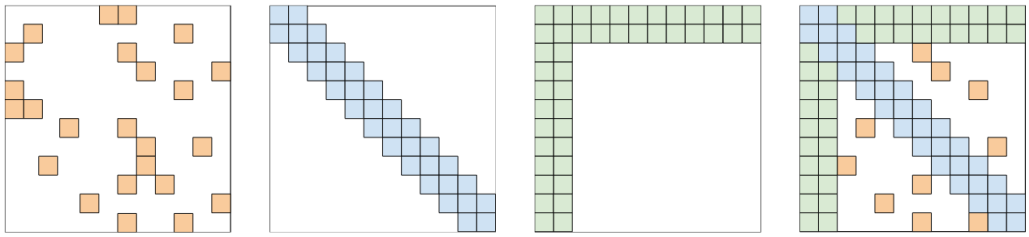


Figure 5.3: The sparse attention mechanism of BigBird [122].

5.3 Proposed ICDBigBird Model

To enhance the performance of previous approaches, we developed ICDBigBird, which we demonstrate can successfully integrate a Graph Convolutional Network (GCN), which takes advantage of the relations between ICD codes, with a BigBird contextual model that can process large documents.

A Graph Convolutional Network (GCN) [51] is a neural network architecture that can capture the general knowledge about the connection between entities. Specifically, GCN builds a symmetric adjacency matrix based on a predefined relationship graph, and the representation of each node is calculated according to its neighbors [51].

In this section, we will describe the proposed ICDBigBird model. First, we will construct the ICD Graph Convolutional Network, which creates an 'enriched' representation

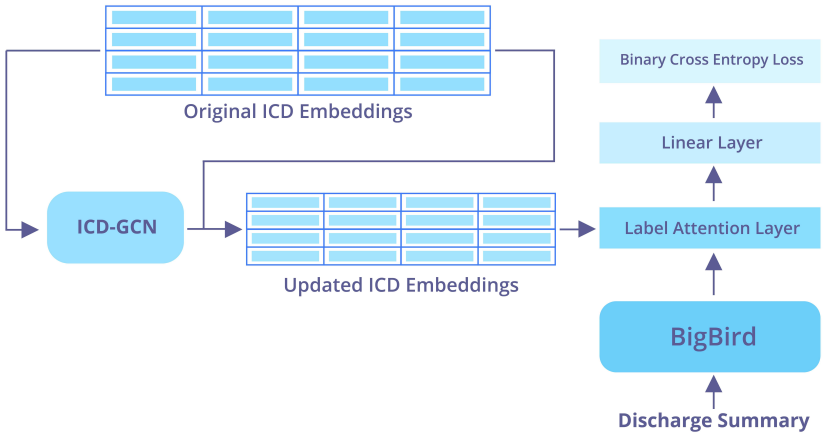


Figure 5.4: ICDBigBird model architecture. ICDBigBird has the ability to integrate a Graph Convolutional Network that takes advantage of the relations between codes with a BigBird contextual model which can process large documents for the ICD classification task.

of the embeddings of ICD codes by taking into account the relation between the different codes. Then, we will describe the BigBird model, which is used for creating the contextual representation of each discharge summary and the label attention layer that showcases the most relevant information to the ICD codes in the representation of each document. An overview of the architecture of ICDBigBird is depicted in Figure 5.4.

5.3.1 ICD Graph Convolutional Network

We use a GCN to capture a more 'enriched' representation for each ICD code. In order to use the ICD-GCN, we first construct an adjacency matrix $A \in \mathbb{R}^{n \times n}$ (where n is the number of unique ICD codes) to represent the connections of ICD codes by using the normalized point-wise mutual information (NPMI) which is described in equation 5.1:

$$NPMI(i,j) = -\frac{1}{\log p(i,j)} \log \frac{p(i,j)}{p(i)p(j)} \tag{5.1}$$

where i and j are different ICD codes and $p(i,j) = \frac{N(i,j)}{N}$, $p(j) = \frac{N(j)}{N}$ where $N(i,j)$ is the number of documents that are labeled with both i and j codes, $N(i)$ is the number of documents that are labeled with i code and N is the total number of documents. We

create an edge between two codes if their NPMI value is greater than a threshold. We empirically set the threshold to 0.2 after experimenting with different threshold values.

We decide to create the adjacency matrix of the ICD-GCN by utilizing the NPMI values instead of considering the hierarchical associations of the ICD codes. This is because we mainly focus on the task of classifying the top 50 most frequent ICD codes (a popular sub-problem of the ICD classification task) [96], where we find little to no hierarchical connection between these codes.

We then construct a definition (sentence) embedding matrix for all the ICD codes using their ICD-9 (sentence) definition from the MIMIC III dataset [47] and the pre-trained sentence transformer embedding model of Reimers et al.[87], as it has been shown to outperform other state-of-the-art sentence embedding methods.

An updated representation of all ICD codes from the ICD-GCN is calculated as follows (equation 5.2) :

$$\hat{U} = Relu(\hat{A}XW) \tag{5.2}$$

where $X \in \mathbb{R}^{n \times m}$ is the definition embedding matrix, n is the number of ICD codes, m is the size of the definition-sentence embedding of each ICD code, $W \in \mathbb{R}^{m \times h}$ is the weight matrix, h is the BigBird's hidden dimension and $\hat{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix where $D_{ii} = \sum_j A_{ij}$.

We concatenate the output of the ICD-GCN with the initial embedding of ICD codes in order to get a richer representation of the codes [88] (equation 5.3):

$$U = \hat{U} \parallel X, U \in \mathbb{R}^{n \times (m+h)} \tag{5.3}$$

5.3.2 ICDBigBird Model

Assuming that a discharge summary has n words, the model's tokenizer generates tokens for each word in the document. Subsequently, the tokens are passed through the input embedding layer of the BigBird model. The input embeddings in turn are passed through multiple attention-based layers, where each layer produces a contextualized embedding of each token. Then, the model produces the final contextual representation of the document $H \in \mathbb{R}^{t \times h}$, where $t = 4096$ is the number of tokens and h is the BigBird's hidden dimension. We use a fully connected linear layer for creating the $\hat{H}$ which is the final representation of the BigBird's embeddings (equation 5.4):

$$\hat{H} = Relu(HW_1) \tag{5.4}$$

where $\hat{H} \in \mathbb{R}^{t\times(m+h)}$ and $W_1 \in \mathbb{R}^{h\times(m+h)}$. Afterwards, we apply a per-label attention mechanism, in order to incorporate the most relevant information to the ICD codes in the contextual representation of each document. Formally, using the $U \in \mathbb{R}^{n\times(m+h)}$ which is the 'updated' ICD coding definition-sentence embeddings matrix, we can compute the dot product attention as (equation 5.5):

$$A = SoftMax(U\hat{H}^{\top}) \tag{5.5}$$

where $A \in \mathbb{R}^{n\times t}$. After calculating the attention score, the output of the attention layer can be calculated as (equation 5.6):

$$V = A\hat{H} \tag{5.6}$$

where $V \in \mathbb{R}^{n\times(m+h)}$. Given the 'updated' representation V, we can compute a probability for each label by using a sum-pooling operation and a sigmoid transformation over the linear projection of V (equation 5.7) :

$$\hat{y} = \sigma(pooling(V \circ W)) \tag{5.7}$$

where $W \in \mathbb{R}^{n\times(m+h)}$.

As the ICD task is a multi-label scenario, the loss function that is typically used is a multi-label binary cross entropy loss (equation 5.8):

$$L_{BCE}(y, \hat{y}) = -\sum_{i=1}^{n}(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)) \tag{5.8}$$

where y is the ground truth label and $\hat{y}$ is a score vector of the ICD codes that our model predicts for each document. However, due to the extremely imbalanced nature of ICD codes, we decide to adopt the Label-Distribution Aware Margin (LDAM) [19]. In the LDAM loss function the output value is subtracted by a label-dependent margin Δ_i before the sigmoid function (equation 5.9):

$$\hat{y}' = \sigma(pooling(V \circ W) - \mathbf{1}(y_i = 1)\Delta_i) \tag{5.9}$$

where $\mathbf{1}(.)$ outputs 1 if y_i=1 and $\Delta_i = \frac{C}{n_i^{1/4}}$ where n_i is number of instances of the i ICD code in the training data and C is a hyper-parameter that needs to be tuned. It should be noted that the main goal of LDAM loss is to regularize more the minority classes than the popular classes so that it can improve the generalization error of minority classes without sacrificing the model's ability to fit the popular classes [19]. Thus we use the $L_{LDAM} = L_{BCE}(y, \hat{y}')$.

5.4 Experiments

5.4.1 Dataset

Following previous research work in the ICD classification task [74, 46, 61], we conduct our experiments on the subset of the English Multiparameter Intelligent Monitoring in Intensive Care III (MIMIC-III) dataset [47] using the top 50 most frequent ICD codes [96]. The MIMIC dataset consists of anonymized electronic medical records in English of over forty-thousand patients admitted to the intensive care units of the Beth Israel Deaconess Medical Center (Boston, MA, USA) between 2001 and 2012.

We extract the free-text discharge summaries and clinical notes from the MIMIC III dataset containing the 50 most frequent ICD codes. We then concatenate the discharge summaries and notes from the same hospitalization admission into one single document. We use the training/validation/testing split from Mullenbach et al. [74] and Li et al. [61] for a fair comparison. The document set size of our subset of MIMIC-III is 8066 for training, 1574 for validation, and 1729 for testing, respectively. Following the preprocessing procedures used in the experiments of the DCAN model [46], the documents are tokenized, with each token converted to lowercase. Any token that does not contain alphabetic characters is removed. Instead of truncating the documents to 2500 words, we set the token size limit to 4096 for our ICDBigBird model. This allows the model to take full advantage of the available information that could be extracted from each document in the dataset. In total, there are 1345 documents comprising more than 2500 words (with a maximum, minimum, and average lengths of 7567, 105 and 1609 words, respectively).

5.4.2 Experimental Setup

In order to address the reproducibility concerns of the NLP community [25], we provide the search strategy and the bound for each hyperparameter: the batch size is set between 32 and 64, and the learning rate is chosen among the values 2e-5, 3e-5 and 5e-5. We set the number of training epochs between 25 and 30 epochs to allow for maximal performance. The best values are chosen based on micro-F1 scores[1] in the validation set. The final hyperparameters selection of our ICDBigBird model is as follows: batch size 32, learning rate 2e-5, trained on 30 epochs, and we empirically set the C hyper-parameter of the LDAM loss to 2. All the contextual embedding models are implemented using the transformers

[1] https://github.com/jamesmullenbach/caml-mimic

library [118] on PyTorch 1.7.1. All experiments are executed on a Tesla K80 with 64GB of system RAM on Ubuntu 18.04.5 LTS.

5.5 Results

5.5.1 Top-50 ICD Classification Task

	AUC-ROC		F1		
Model	Macro	Micro	Macro	Micro	*P*@5
Att. LSTM [96]	-	90.0	-	53.2	-
BI-GRU [74]	82.8	86.8	48.4	54.9	-
CAML [74]	87.5	90.9	53.2	61.4	-
DRC.[74]	88.4	91.6	57.6	63.3	61.8
LEAM [114]	88.1	91.2	54.0	61.9	61.2
HyperCore [20]	89.5±0.3	92.9± 0.2	60.9 ± 0.1	66.3 ± 0.1	63.2 ± 0.2
Mult.CNN [61]	89.9± 0.4	92.8 ± 0.2	60.6±1.1	67.0±0.3	64.1±0.1
DCAN [46]	**90.2±0.6**	**93.1±0.1**	61.5±0.7	67.1±0.1	64.2±0.2
ICDBigBird	90.0±0.5	92.9 ±0.2	**63.1±0.5**	**69.6±0.1**	**65.4±0.1**
ICDBigB.(val.)	91.0±0.6	93.3 ±0.1	64.1±0.4	70.4±0.1	65.1±0.3
Ablation Study					
BERT[23]	80.3±0.4	84.4 ±0.5	43.7±0.2	51.4±0.5	51.9±0.3
BioBERT[58]	81.3±0.5	85.5 ±0.4	46.3±0.3	54.6±0.3	54.2 ±0.4
C.B.[4]	81.7±0.4	85.8 ±0.5	46.4±0.3	54.3 ±0.4	53.2±0.4
No attention	86.7±0.5	90.4 ±0.3	55.2±0.4	64.8±0.2	62.5±0.3
L. Attention	88.4±0.5	91.2 ±0.2	60.2±0.2	67.8±0.3	63.6±0.5
R. embedding	89.2±0.4	91.8 ±0.5	60.8±0.2	67.8±0.2	63.2±0.1

Table 5.1: Results of mean ± standard deviation of three runs of the ICDBigBird model on the test split of the MIMIC-III dataset with top 50 ICD codes; We also provide the performance of previous state-of-the-art models using the same test set. *C.B.* is the Bio_ClinicalBert; *DRC.* is DR-CAML; *L. Attention* is Linear Attention; *R. embedding* is the random embedding; we also include the results on the validation split of MIMIC III (ICDBigB.(val.)); Best values on the **test** set are **bolded**.

We benchmark our ICDBigBird model against existing state-of-the-art models for the top 50 ICD classification tasks. For all the models, we evaluate the micro and macro aver-

aging F1 score, the receiver operating characteristic curve (AUC-ROC), and the precision at k codes with k=5 (P@5). In Table 5.1, we can observe that our model outperforms all other models in the micro and macro averaging F1 and in the P@5 score with comparable performance on the other two metrics (with the DCAN model [46] achieving the best AUC-ROC results).

5.5.2 Ablation Study

In order to evaluate the effect of each feature on the performance of ICDBigBird, we conduct an ablation study. The results are presented in Table 5.1.

1. We investigate whether the ability of the BigBird model to process large documents can boost the performance of our framework. It can be observed that contextual model architectures that can process documents of at most 512 tokens (Bert [23], Biobert [58], and Bio_ClinicalBert [4]) cannot achieve the performance of a BigBird architecture even if these models are pre-trained on medical documents (BioBert and Bio_ClinicalBert).

2. We examine the effect of the GCN model by testing the performance of contextual embeddings without enriching them with information from the definitions of the codes through an attention mechanism (BigBird without attention) and by substituting the GCN attention mechanism with the typical linear attention mechanism (Linear Attention) [74]. It can be observed that our model benefits from the attention mechanism, as it cannot achieve optimal performance without it. Also, the fact that the GCN graph attention mechanism achieves a better performance than a typical linear attention mechanism is a strong indication that the GCN can provide valuable information about the connections between the ICD codes.

3. Previous research work [46] used a random initialization of the embeddings of the ICD codes. However, in our experiments, a model with random initialization of the embedding of the codes (R. embedding) results in sub-optimal performance. Thus we can conclude that using information from the definitions of the codes to initialize their embeddings has a positive effect on the model's performance.

5.6 Conclusion

In this chapter, we presented a novel contextual model that has the ability to integrate a Graph Convolutional Network model that takes advantage of the relations between codes with a BigBird contextual model which can process large documents for the ICD classification task.

Our evaluation showed the model's efficiency, as it outperformed previous state-of-the-art models for this task by at least 1.5 F1 points. Our ablation study demonstrated that a contextual model that could process large documents (e.g., BigBird) performed better than those that are limited to processing documents with length of, at most, 512 tokens, even if they were trained on medical datasets. We illustrated that using information from the definitions of the codes to initialize the definition-embeddings of the labels had a positive effect on the model's performance. We also showed that the relational information from a Graph Convolutional Network was beneficial for the model's performance.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The digital transformation of our society is creating a tremendous amount of data at an unprecedented rate. A large portion of these data is in unstructured text format [11]. There is an increased interest in NLP tools that can assist us in navigating, understanding, and summarizing useful information from these textual data. This thesis illustrates how structured external knowledge can direct an NLP model to learn the associations between distinctive terminologies, which it otherwise may not have the opportunity to learn, due to the scarcity of domain-specific datasets. By injecting structured domain-specific knowledge into a deep learning NLP architecture, we show that we can to tackle the low-resource NLP challenge, particularly in the biomedical domain. We also demonstrate the effectiveness of these strategies in improving the performance and generalization of contextual models.

Four distinct but complementary strategies are pursued (Figure 6.1). The first strategy is to augment contextual models with structured, general lexical information to aid the model in distinguishing between semantically similar words. In Chapter 2, we present LexSubCon, an end-to-end lexical substitution framework based on contextual embedding models. LexSubCon updates the input policy of a contextual model by introducing a new mix-up embedding strategy for the input embedding of the target word. We also combine features from contextual embedding models and external lexical knowledge bases. These features allow the model to determine the most appropriate substitution words without modifying the meaning of the original sentence, by introducing a new gloss (definition) similarity metric, which calculates the similarity of the sentence-definition embeddings of the target word and its proposed candidates. We confirm that these features can improve

the model's performance, causing it to outperform other state-of-the-art models on two benchmark datasets in the lexical substitution task.

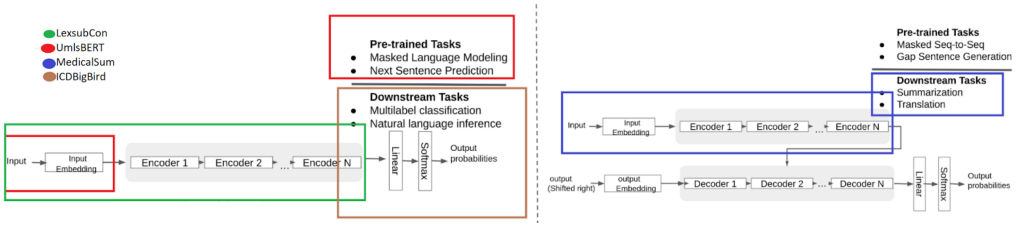


Figure 6.1: Four distinct strategies that are pursued in the thesis.

Having successfully proven that general lexical structured knowledge can aid a contextual model in distinguishing between semantically similar words, we extend this exploration by incorporating domain-specific knowledge in the pre-training process of a contextual word embeddings model (BERT). In Chapter 3, we focus our investigation in the biomedical field using a standardized domain knowledge base, namely UMLS, to demonstrate the effectiveness of our strategy. Our model, UmlsBERT, introduces domain knowledge in the pre-training process and architecture of a BERT model. We propose a new multi-label loss function for the Masked Language Modelling (Masked LM) pre-training task that takes into consideration the connection between words that share the same concept unique identifier (CUI) attribute in UMLS. We also introduce a semantic type embedding that enriches the input embedding process by leveraging information from the semantic type of each biomedical word. We demonstrate that the augmented model, namely UmlsBERT, can learn the association of different clinical terms with similar meaning in the UMLS Metathesaurus and the association between words of the same semantic type. We confirm that these strategies can improve the model's performance, as UmlsBERT outperforms other biomedical BERT models in various medical downstream tasks (e.g., named-entity recognition (NER) and clinical natural language inference).

We further extend the usage of medical knowledge in UMLS and we illustrate that structured medical knowledge can also boost the performance of a (medical) summarization transformer-based sequence-to-sequence model. We demonstrate that injecting structured medical knowledge into a sequence-to-sequence summarization model can aid the model in including relevant medical facts in the summarized output. In Chapter 4, we propose MedicalSum, an architecture that takes advantage of external medical knowledge to identify key medical information in the input text. MedicalSum uses a novel weighted loss function that incentivizes the model to predict words with a medical meaning. MedicalSum also creates more meaningful input embeddings insofar as it can force the embedding of the

words of the same semantic type to become more similar in the embedding space by incorporating information from the semantic type of each biomedical word. Our model outperforms the baseline model on four medical summarization datasets. Our qualitative analysis also shows that these features can guide the summarization process to include relevant medical facts in the summarized output.

We also tackle a challenging NLP downstream task in the biomedical field, the multi-label classification problem of automatic code assignment. We demonstrate the benefit of knowledge augmentation in the ICD coding task in Chapter 5. We propose ICDBig-Bird, an architecture that integrates a Graph Convolutional Network (GCN), which takes advantage of the relations between ICD codes, with a BigBird contextual model that can process large documents. Experiments on the MIMIC III dataset confirm that ICDBigBird outperforms the existing state-of-the-art models. Furthermore, the ablation study shows that the information from a GCN (created by taking into consideration the normalized point-wise mutual information of the ICD codes in the medical documents) is beneficial to the model's performance.

In summary, this thesis examined knowledge augmentation strategies in every component of transformer-based NLP models and different downstream tasks such as lexical substitution, entity recognition, medical summarization, and multi-label classification. We proposed novel strategies for injecting structured domain knowledge (from general-purpose lexical knowledge to biomedical domain-specific knowledge bases) into contextual models to tackle the low-resource NLP challenge and demonstrated their positive impact on model performance. With a focus on the biomedical domain, the models proposed in this thesis have the potential to accelerate the adoption of NLP tools in clinical research and practice.

6.2 Future Work

The scarcity of domain-specific datasets poses significant challenges in utilizing high-performing, optimized NLP models. This dissertation aspires to make a few steps towards injecting domain-specific knowledge into contextual models to tackle the low-resource NLP challenge, particularly in the biomedical domain. Our approach to this vision, which is reflected in a similar pattern underlying this thesis, starts with understanding the challenges in each NLP task and then explores novel solutions to specific problems. We believe that some natural extensions to this dissertation would be the following:

Augmenting Contextual Models with General Lexical Knowledge: Lexical Substitution is the task of generating appropriate words which can replace a target word in

a given sentence without changing the sentence's meaning. The increased research interest in Lexical Substitution is due to its utility in various NLP fields such as data augmentation and paraphrase generation. The process of incorporating structured knowledge into the architecture of a contextual model in order to aid the model in distinguishing which words are semantically similar presents an exciting spectrum of research opportunities. We plan to pursue the following directions: (i) investigating novel methods to encode the knowledge on asymmetric relations such as meronymy [57]; (ii) exploring other features for ranking the candidates (e.g. parser information [102]); and (iii) testing the strategies mentioned above on datasets in other languages using multi-language lexical databases (e.g., MultiWordNet [83] or BalkaNet [108]) to investigate whether these features could have the same effect on different languages.

Augmentation of Contextual Models in the Biomedical Domain: Contextual word embedding models have achieved state-of-the-art results in many (clinical) NLP tasks such as entity recognition and biomedical question answering [23, 58]. Augmenting a transformer-based encoder model with structured knowledge from a specific (medical) domain can aid the model in learning more easily the associations between distinctive terminologies. We want to expand the techniques that we designed for augmenting contextual models with structured medical information in the following directions: i) analyzing how the model's performance can be affected by UMLS hierarchical associations between words (e.g. from general to more specific concepts accompanied by non-essential modifiers) that extend the concept connection investigated in this dissertation; (ii) examining the effect of augmenting contextual embeddings with medical knowledge when more complicated layers are used atop the output embedding of the medical knowledge-augmented contextual architecture; (iii) testing the medical-augmented contextual models in other biomedical tasks (e.g. relation extraction task [53]) to further investigate the effect of structured medical knowledge on the performance of a contextual model.

Augmenting Transformer-based sequence-to-sequence model for Summarizing Medical Conversations: Traditionally, clinical professionals review clinical documents and manually create the appropriate summaries by following specific guidelines. Medical note generation by abstractive summarization can be used to automate clinical documentation and reduce the workload associated with creating summaries of clinical encounters. Providing external knowledge in the summarization decision process presents a wide range of open challenges and opportunities. In particular, we plan to tackle three critical aspects: (i) examining different guidance signals, such as the inclusion of relational triples between medical entities (e.g. a relation between disease and its symptoms); (ii) testing different evaluation metrics for summarization, as the ROUGE metric may not be the most suitable for summarization evaluation, especially in summaries with high termi-

nology variations [22]; (iii) investigating the effects of guiding the summarization process with soft templates for the case of real-world hospital summarization where each hospital has a specific note template.

Contextual models for Imbalanced Multi-Label Classification Problems in the Biomedical Domain: The ICD coding task is crucial for making clinical, operational, and financial decisions in healthcare. Traditionally, medical coders review clinical documents and manually assign the appropriate ICD codes by following specific coding guidelines. Automatic coding classification could help save time and cost in data extraction and reporting. We plan to extend our studies of integrating relational information with contextual models in the following directions: (i) investigating the effects of using the hierarchical structure of ICD codes for strengthening the GCN embeddings (especially in the case of the complete ICD code set); (ii) testing the generalizability of an ICD classification model that is trained on the complete ICD code set when it is provided with datasets from a specific medical subdomain (e.g. family medicine or cardiac) that only contain the medical codes of their respective field.

References

[1] Griffin Adams, Emily Alsentzer, Mert Ketenci, Jason Zucker, and Noémie Elhadad. What's in a summary? laying the groundwork for advances in hospital-course summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4794–4811, Online, June 2021. Association for Computational Linguistics.

[2] Rodrigo Agerri, Xabier Artola, Zuhaitz Beloki, German Rigau, and Aitor Soroa. Big data for natural language processing: A streaming approach. *Knowledge-Based Systems*, 79:36–42, 2015.

[3] Milam Aiken and Mina Park. The efficacy of round-trip translation for mt evaluation. *Translation Journal*, 14, 02 2010.

[4] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[5] Nikolay Arefyev, Boris Sheludko, Alexander Podolskiy, and Alexander Panchenko. A comparative study of lexical substitution approaches based on neural language models. *arXiv preprint arXiv:2006.00031*, 2020.

[6] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations (ICLR)*, 2017.

[7] Segun Taofeek Aroyehun and Alexander Gelbukh. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In

Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), pages 90–97, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[8] Anand Avati, Kenneth Jung, Stephanie Harman, Lance Downing, A. Ng, and Nigam Haresh Shah. Improving palliative care with deep learning. *BMC Medical Informatics and Decision Making*, 18, 2018.

[9] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[10] Dzmitry Bahdanau, Kyunghyun Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, 09 2014.

[11] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33, Toulouse, France, July 2001. Association for Computational Linguistics.

[12] Atreya Basu, Carolyn Watters, and Michael Author. Support vector machines for text categorization. page 103, 01 2003.

[13] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. A survey on data augmentation for text classification. *ArXiv*, abs/2107.03158, 2021.

[14] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: Pretrained language model for scientific text. In *EMNLP*, 2019.

[15] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.

[16] Asma Ben Abacha, Chaitanya Shivade, and Dina Demner-Fushman. Overview of the MEDIQA 2019 shared task on textual inference, question entailment and question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 370–379, Florence, Italy, August 2019. Association for Computational Linguistics.

[17] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[18] O. Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res.*, 32(Database issue):D267–270, Jan 2004.

[19] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019.

[20] Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. HyperCore: Hyperbolic and co-graph representation for automatic ICD coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3105–3114, Online, July 2020. Association for Computational Linguistics.

[21] Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California, June 2016. Association for Computational Linguistics.

[22] Arman Cohan and Nazli Goharian. Revisiting summarization evaluation for scientific articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 806–813, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[24] Ish Kumar Dhammi and Sudhir Kumar. Medical subject headings (mesh) terms. *Indian journal of orthopaedics vol.*, 48,5, 2014.

[25] Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. Show your work: Improved reporting of experimental results. In *Proceedings of EMNLP*, 2019.

[26] Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online, June 2021. Association for Computational Linguistics.

[27] Seppo Enarvi, Marilisa Amoia, Miguel Del-Agua Teba, B. Delaney, Frank Diehl, Stefan Hahn, Kristina Harris, Liam R. McGrath, Yue Pan, Joel Pinto, Luca Rubini, Miguel Ruiz, Gagandeep Singh, Fabian Stemmer, Weiyi Sun, Paul Vozila, Thomas Lin, and Ranjani Ramamurthy. Generating medical reports from patient-doctor conversations using sequence-to-sequence models. In *NLPMC*, 2020.

[28] Katrin Erk and Sebastian Padó. Exemplar-based models for word meaning in context. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 92–97, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[29] Gregory Finley, Erik Edwards, Amanda Robinson, Michael Brenndoerfer, Najmeh Sadoughi, James Fone, Nico Axtmann, Mark Miller, and David Suendermann-Oeft. An automated medical scribe for documenting clinical encounters. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–15, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[30] Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China, November 2019. Association for Computational Linguistics.

[31] Aina Garí Soler, Anne Cocos, Marianna Apidianaki, and Chris Callison-Burch. A comparison of context-sensitive models for lexical substitution. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 271–282, Gothenburg, Sweden, May 2019. Association for Computational Linguistics.

[32] Claudio Giuliano, Alfio Gliozzo, and Carlo Strapparava. FBK-irst: Lexical substitution task exploiting domain and syntagmatic coherence. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 145–148, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[33] 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, 2020.

[34] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48, 07 2017.

[35] Boran Hao, Henghui Zhu, and Ioannis Paschalidis. Enhancing clinical BERT embedding using a biomedical knowledge base. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 657–661, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[36] Samer Hassan, Andras Csomai, Carmen Banea, Ravi Sinha, and Rada Mihalcea. UNT: SubFinder: Combining knowledge sources for automatic lexical substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 410–413, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[37] Yun He, Ziwei Zhu, Yin Zhang, Qin Chen, and James Caverlee. Infusing Disease Knowledge into BERT for Health Question Answering, Medical Inference and Disease Name Recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4604–4614, Online, November 2020. Association for Computational Linguistics.

[38] Marti Hearst, S.T. Dumais, E. Osman, John Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13:18 – 28, 08 1998.

[39] Gerold Hintz and Chris Biemann. Language transfer learning for supervised lexical substitution. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 118–129, Berlin, Germany, August 2016. Association for Computational Linguistics.

[40] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[41] Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3507–3512, Hong Kong, China, November 2019. Association for Computational Linguistics.

[42] William J. Hutchins. Machine translation: A brief history, 1995.

[43] International classification of diseases,ninth revision, clinical modification (icd-9-cm). https://www.cdc.gov/nchs/icd/icd9cm.htm/, 2022. visited on 2022-05-21.

[44] Olaronke G Iroju and Janet O Olaleke. A systematic review of natural language processing in healthcare. *International Journal of Information Technology and Computer Science*, 8:44–50, 2015.

[45] Serena Jeblee, Faiza Khan Khattak, Noah Crampton, Muhammad Mamdani, and Frank Rudzicz. Extracting relevant information from physician-patient dialogues for automated clinical note taking. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 65–74, Hong Kong, November 2019. Association for Computational Linguistics.

[46] Shaoxiong Ji, Erik Cambria, and Pekka Marttinen. Dilated convolutional attention network for medical code assignment from clinical text. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 73–78, Online, November 2020. Association for Computational Linguistics.

[47] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[48] Karen Sparck Jones. Natural language processing: a historical review. In *Current Issues in Computational Linguistics: in Honour of Don Walker (Ed Zampolli, Calzolari and*. Kluwer, 1994.

[49] Anirudh Joshi, Namit Katariya, X. Amatriain, and Anitha Kannan. Dr. summarize: Global summarization of medical dialogue by exploiting local structures. In *Findings of the Association for Computational Lingustics: EMNLP 2020*, 2020.

[50] Rajvir Kaur and Jeewani Anupama Ginige. Comparative analysis of algorithmic approaches for auto-coding with icd-10-am and achi. *Studies in health technology and informatics*, 252:73–79, 2018.

[51] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, 2017.

[52] Zeljko Kraljevic, Thomas Searle, Anthony Shek, Lukasz Roguski, Kawsar Noor, Daniel Bean, Aurelie Mascio, Leilei Zhu, Amos A Folarin, Angus Roberts, Rebecca Bendayan, Mark P Richardson, Robert Stewart, Anoop D Shah, Wai Keong Wong, Zina Ibrahim, James T Teo, and Richard J B Dobson. Multi-domain clinical natural

language processing with MedCAT: The medical concept annotation toolkit. *Artif. Intell. Med.*, 117:102083, July 2021.

[53] Martin Krallinger, O. Rabal, S. A. Akhondi, M. Pérez, Jesús Santamaría, Gael Pérez Rodríguez, G. Tsatsaronis, Ander Intxaurrondo, J. A. López, Umesh Nandal, E. V. Buel, A. Chandrasekhar, Marleen Rodenburg, A. Lægreid, Marius A. Doornenbal, J. Oyarzábal, Anália Lourenço, and A. Valencia. Overview of the biocreative vi chemical-protein interaction track. In *In Proceedings of the sixth BioCreative challenge evaluation workshop*, pages 141—146, 2017.

[54] Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. What substitutes tell us - analysis of an "all-words" lexical substitution corpus. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 540–549, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[55] Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, page 68–73, New York, NY, USA, 1995. Association for Computing Machinery.

[56] Ronilda C Lacson, Regina Barzilay, and William J Long. Automatic analysis of medical dialogue in the home hemodialysis domain: structure induction and summarization. *Journal of biomedical informatics*, 39(5):541–555, 2006.

[57] Anne Lauscher, Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. Specializing unsupervised pretraining models for word-level semantic similarity. 2019.

[58] 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, 09 2019.

[59] Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. SenseBERT: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online, July 2020. Association for Computational Linguistics.

[60] Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. Guiding generation for abstractive text summarization based on key information guide network. In *Proceedings of the*

2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 55–60, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[61] Fei Li and Hong Yu. Icd coding from clinical text using multi-filter residual convolutional neural network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8180–8187, 04 2020.

[62] Daniel Loureiro and Alípio Jorge. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy, July 2019. Association for Computational Linguistics.

[63] Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8449–8456, Apr. 2020.

[64] Alexandre Magueresse, Vincent Carles, and Evan Heetderks. Low-resource languages: A review of past work and future challenges, 2020.

[65] David Martinez, Su Nam Kim, and Timothy Baldwin. MELB-MKB: Lexical substitution system based on relatives in context. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 237–240, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[66] Diana McCarthy and Roberto Navigli. SemEval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[67] Alexa McCray, Anita Burgun, and Olivier Bodenreider. Aggregating umls semantic types for reducing conceptual complexity. *Studies in health technology and informatics*, 84:216–20, 02 2001.

[68] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018.

[69] Oren Melamud, Ido Dagan, and Jacob Goldberger. Modeling word meaning in context with substitute vectors. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 472–482, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[70] Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany, August 2016. Association for Computational Linguistics.

[71] Oren Melamud, Omer Levy, and Ido Dagan. A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, Denver, Colorado, June 2015. Association for Computational Linguistics.

[72] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[73] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[74] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1101–1111, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[75] Nidhi and G Vishal. Recent trends in text classification techniques. *International Journal of Computer Applications*, 35:45–51, 2011.

[76] World Health Organization. Icd-10 : international statistical classification of diseases and related health problems : tenth revision, 2004.

[77] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[78] Naima Oubenali, Sabrina Messaoud, Alexandre Filiot, Antoine Lamer, and Paul Andrey. Visualization of medical concepts represented using word embeddings: a scoping review. *BMC Medical Informatics and Decision Making*, 22, 03 2022.

[79] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, page 271–es, USA, 2004. Association for Computational Linguistics.

[80] Thomas H. Payne, W. David Alonso, J. Andrew Markiel, Kevin Lybarger, and Andrew A. White. Using voice to create hospital progress notes: Description of a mobile application and supporting system integrated with a commercial electronic health record. *Journal of Biomedical Informatics*, 77:91–96, 2018.

[81] Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65, 2019.

[82] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[83] Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. Multiwordnet: developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet*, January 2002.

[84] Rimma Pivovarov and Noémie Elhadad. Automated methods for the summarization of electronic health records. *Journal of the American Medical Informatics Association : JAMIA*, 22, 04 2015.

[85] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[86] Lance Ramshaw and Mitch Marcus. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*, 1995.

[87] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[88] Anthony Rios and Ramakanth Kavuluru. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3132–3142, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[89] Stephen Roller and Katrin Erk. PIC a different word: A simple model for lexical substitution in context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1121–1126, San Diego, California, June 2016. Association for Computational Linguistics.

[90] Stephen Roller and Katrin Erk. PIC a different word: A simple model for lexical substitution in context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1121–1126, San Diego, California, June 2016. Association for Computational Linguistics.

[91] Alexey Romanov and Chaitanya Shivade. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1586–1596, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[92] Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[93] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

[94] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics, 2017.

[95] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics.

[96] Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P. Xing. Towards automated icd coding using deep learning. *arXiv preprint arXiv:1711.04075*, 2017.

[97] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[98] Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1. *Journal of biomedical informatics*, 58 Suppl:S11–9, 2015.

[99] Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association : JAMIA*, 20, 04 2013.

[100] Simon Šuster, Stéphan Tulkens, and Walter Daelemans. A short review of ethical challenges in clinical natural language processing. *arXiv preprint arXiv:1703.10090*, 2017.

[101] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[102] György Szarvas, Chris Biemann, and Iryna Gurevych. Supervised all-words lexical substitution using delexicalized features. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, pages 1131–1141, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[103] Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 948–957, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[104] The next 10 years look golden for natural language processing research. https://www.microsoft.com/en-us/research/lab/microsoft-research-asia/articles/next-10-years-natural-language-processing/, 2022. visited on 2022-04-15.

[105] The web's free icd-9-cm medical coding reference. http://www.icd9data.com/2015/Volume1/320-389/320-327/320/default.htm, 2022. visited on 2022-03-27.

[106] Jörg Tiedemann and Santhosh Thottingal. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

[107] Brian D Tran, Yunan Chen, Songzi Liu, and Kai Zheng. How does medical scribes' work inform development of speech-based clinical documentation technologies? A systematic review. *Journal of the American Medical Informatics Association*, 27(5):808–817, 03 2020.

[108] D. Tufis, D. Cristea, and S. Stamou. Balkanet: Aims, methods, results and perspectives. a general overview. *In: D. Tufiş (ed): Special Issue on BalkaNet. Romanian Journal on Science and Technology of Information*, pages 3–4, 2004.

[109] Understanding lstm networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015. visited on 2022-05-21.

[110] Ozlem Uzuner, Yuan Luo, and Peter Szolovits. Evaluating the state-of-the-art in automatic de-identification. *Journal of the American Medical Informatics Association : JAMIA*, 14:550–63, 06 2007.

[111] Ozlem Uzuner, Brett South, Shuying Shen, and Scott DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association : JAMIA*, 18:552–6, 06 2011.

[112] M. M. van Buchem, H. Boosman, M. P. Bauer, I. Kant, S. Cammel, and E. Steyerberg. The digital scribe in clinical practice: a scoping review and research agenda. *NPJ Digital Medicine*, 4, 2021.

[113] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017.

[114] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2321–2331, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[115] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

[116] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China, November 2019. Association for Computational Linguistics.

[117] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018.

[118] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[119] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick,

Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144, 2016.

[120] Jian Yang, Gang Xiao, Yulong Shen, Wei Jiang, Xinyu Hu, Ying Zhang, and Jinghui Peng. A survey of knowledge enhanced pre-trained models. *ArXiv*, abs/2110.00269, 2021.

[121] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13:55–75, 08 2018.

[122] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33, 2020.

[123] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[124] Yuhao Zhang, Daisy Yi Ding, Tianpei Qian, Christopher D. Manning, and Curtis P. Langlotz. Learning to summarize radiology findings. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 204–213, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[125] Zachariah Zhang, Jingshu Liu, and Narges Razavian. BERT-XML: Large scale automated ICD coding using BERT pretraining. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 24–34, Online, November 2020. Association for Computational Linguistics.

[126] Jianing Zhou and Suma Bhat. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[127] Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. BERT-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy, July 2019. Association for Computational Linguistics.

[128] Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. Enhancing factual consistency of abstractive summarization. *arXiv preprint arXiv:2003.08612*, 2020.

APPENDICES

Appendix A

IOB Format

The IOB format (short for Inside-Outside-Beginning) was defined for the CoNLL-2003 shared task on the named-entity recognition (NER) task. In the IOB format, every word in each chunk can be categorized with one of the following three labels:

- **I-:** The I- prefix indicates that the word is inside a chunk.
- **O-:** The O- prefix indicates that the word is a token that belongs to no chunk (outside of any chunk).
- **B-:** The B- prefix indicates that the word is at the beginning of a chunk.

  For example, given the sentence:

  His sister stated that the mother had a progressive mental decline.

  with NER Label:

  "NULL NULL EVIDENTIAL NULL NULL NULL PROBLEM PROBLEM PROBLEM'

  its IOB format would be:

"O O B-EVIDENTIAL O O O B-PROBLEM I-PROBLEM I-PROBLEM'

Appendix B

UMLS Metathesaurus

Table B.1 is a list of all semantic types and their semantic groups used in the UmlsBERT architecture.

Semantic Group	Semantic Type
Chemicals & Drugs	Amino Acid, Peptide, or Protein
Disorders	Acquired Abnormality
Disorders	Anatomical Abnormality
Chemicals &Drugs	Biologically Active Substance
Anatomy	Body System
Anatomy	Body Location or Region
Chemicals &Drugs	Biomedical or Dental Material
Anatomy	Body Part, Organ, or Organ Component
Anatomy	Body Space or Junction
Anatomy	Cell Component
Physiology	Cell Function
Anatomy	Cell
Disorders	Congenital Abnormality

Disorders	Cell or Molecular Dysfunction
Procedures	Diagnostic Procedure
Activities &Behaviors	Daily or Recreational Activity
Disorders	Disease or Syndrome
Chemical &Drugs	Element, Ion or Isotope
Chemicals&Drugs	Enzyme
Disorders	Finding
Chemicals &Drugs	Hazardous or Poisonous Substance
Physiology	Genetic Function
Chemicals&Drugs	Hormone
Chemicals&Drugs	Immunologic Factor
Chemicals&Drugs	Inorganic Chemical
Disorders	Injury or Poisoning
Chemicals&Drugs	Indicator, Reagent, or Diagnostic Aid
Procedures	Laboratory Procedure
Physiology	Mental Process
Disorders	Mental or Behavioral Dysfunction
Physiology	Molecular Function
Disorders	Neoplastic Process
Activities&Behaviors	Occupational Activity
Chemicals&Drugs	Organic Chemical
Physiology	Organism Function
Physiology	Organ or Tissue Function
Disorders	Pathologic Function
Chemicals&Drugs	Pharmacologic Substance
Disorders	Sign or Symptom
Anatomy	Tissue
Chemical&Drugs	Vitamin

Table B.1: Example of Semantic Types and Semantic Group names that are used in the UmlsBERT architecture