UNICA

UNIVERSITÀ
DEGLI STUDI
DI CAGLIARI

Ph.D. DEGREE IN
MATHEMATICS AND COMPUTER SCIENCE

Cycle XXXV

TITLE OF THE Ph.D. THESIS

Knowledge Augmentation in Language Models to Overcome Domain Adaptation and Scarce Data Challenges in Clinical Domain

Scientific Disciplinary Sector(s)

INF/01 INFORMATICA

Ph.D. Student:	Vivek Kumar
Supervisor	Prof. Diego Reforgiato Recupero (UniCa)
Co-Supervisor	Prof. Daniele Riboni (UniCa)

Final exam. Academic Year 2021/2022
Thesis efence: April 2023 Session

Abstract

The co-existence of two scenarios, "the massive amount of unstructured text data that humanity produces" and "the scarcity of sufficient training data to train language models," in the healthcare domain have multifold increased the need for intelligent tools and techniques to process, interpret and extract different types of knowledge from the data. My research goal in this thesis is to develop intelligent methods and models to automatically better interpret human language and sentiments, particularly its structure and semantics, to solve multiple higher-level Natural Language Processing (NLP) downstream tasks and beyond.

This thesis is spread over six chapters and is divided into two parts based on the contributions. The first part is centered on best practices for modeling data and injecting domain knowledge to enrich data semantics applied to tackle several classification tasks in the healthcare domain and beyond. The contribution is to reduce the training time, improve the performance of classification models, and use world knowledge as a source of domain knowledge when working with limited/small training data. The second part introduces the one of its kind high-quality dataset of Motivational Interviewing (MI), AnnoMI, followed by the experimental benchmarking analysis for AnnoMI. The contribution accounts to provide a publicly accessible dataset of Motivational Interviewing and methods to overcome data scarcity challenges in complex domains (such as mental health). The overall organization of the thesis is as follows:

The first chapter provides a high-level introduction to the tools and techniques applied in the scope of the thesis.

The second chapter presents optimal methods for (i) feature selection, (ii) eliminating irrelevant and superfluous attributes from the dataset, (iii) data preprocessing, and (iv) advanced data representation methods (word embedding and bag-of-words) to model data.

The third chapter introduces the Language Model (LM), K-LM, a combination of

Generative Pretrained Transformer (GPT)-2 and Bidirectional Encoder Representations from Transformers (BERT) that uses knowledge graphs to inject domain knowledge for domain adaptation tasks. The end goal of this chapter is to reduce the training time and improve the performance of classification models when working with limited/small training data.

The fourth chapter introduces the high-quality dataset of expert-annotated MI (AnnoMI), comprised of 133 therapy session transcriptions distributed over 44 topics (including smoking cessation, anxiety management, weight loss, etc.), and provides an in-depth analysis of the dataset.

The fifth chapter presents the experimental analysis with AnnoMI, which includes (i) augmentation techniques to generate data and (ii) fairness and bias assessments of the employed Classical Machine Learning (CML) and Deep Learning (DL) approach to develop reliable classification models.

Finally, the sixth chapter provides the conclusion and outcomes of all the work presented in this thesis. The scientific contributions of this thesis include the solution to overcome the challenges of scarce training data in complex domains and domain adaptation in LMs. The practical contributions of the thesis are data resources and the language model for a range of quantitative and qualitative NLP applications.

Keywords: Natural Language Processing, Domain Adaptation, Motivational Interviewing, AI Fairness and Bias, Data Augmentation, GPT, BERT, Healthcare.

Acknowledgement

A Ph.D. is a long learning path and I am thankful for the many people I got to meet and who accompanied me in this way. First, and foremost I would like to thank my project supervisors Prof. Dr. Diego Reforgiato Recupero and Prof. Dr. Daniele Riboni- my academic supervisors at *UniCa* and Dr. Rim Helaoui- my industrial supervisor at *Philips Research*, for their continuous support and guidance throughout the study and research. I am grateful for their patience, motivation, enthusiasm, kindness, and immense knowledge. I could not have imagined having better advisors and mentors for my Ph.D. I owe you my scientific accomplishments.

My sincere thanks also go to my reviewers, Prof. Dr. Giovanni Farinella and Prof. Dr. Stefano Montanelli, for their valuable feedbacks.

I thank my fellow researchers Simone Balloccu and Zixiu Wu for the productive collaborations, stimulating discussions, and sleepless nights we were working to meet the deadlines.

I would like to thank the European Commission's Horizon 2020, Marie Sklodowska-Curie Actions, University of Cagliari, Philips Research and beneficiaries associated with Personal Health Interfaces Leveraging Human–Machine Natural Interactions (PhilHumans) (grant number 812882) under which my work is concluded.

Finally, I wish to thank Lord Mahakal and my parents and family for their support and encouragement throughout this journey.

कर्मण्येवाधिकारस्ते मा फलेषु कदाचन ।
मा कर्मफलहेतुर्भूर्मा ते सङ्गोऽस्त्वकर्मणि ।।

- भगवान् श्रीकृष्ण -
श्रीमद् भगवद्गीता : अध्याय 2, श्लोक 47

|| karmaṇy-evādhikāras te mā phaleṣhu kadāchana,
mā karma-phala-hetur bhūr mā te saṅgo 'stvakarmaṇi ||

Thy right is to work only,
but never with its fruits;
let not the fruits of action be thy motive,
nor let thy attachment be to inaction.

Tu hai il diritto di compiere i tuoi doveri prescritti,
ma non di godere dei frutti dell'azione.
Non credere mai di essere la causa delle conseguenze dell'azione,
e non cercare mai di sfuggire al tuo dovere.

- Lord Shri. Krishna -
Bhagavad Gita: Chapter 2 | Verse 47

Preface

This thesis is the culmination of my research works done from 2019 to 2002 at the *University of Cagliari* (UniCA), Italy and *Philips Research*, Eindhoven, Netherlands. The scientific work in this thesis is in accordance with the goals of the international framework of **PhilHumans** (Personal Health Interfaces Leveraging Human-Machine Natural Interactions) project. **PhilHumans** (Grant Agreement ID: 812882) is an ambitious project under **Marie Skłodowska-Curie Actions** (MSCA) of the **European Union's Horizon 2020** Innovative Training Networks (ITN) program.

Marie Skłodowska-Curie Actions is a set of major research fellowships created by the European Union/European Commission to support research in the European Research Area. The Marie Skłodowska-Curie Actions are among Europe's most competitive and prestigious research and innovation fellowships.

The **PhilHumans** consortium has a generous budget of € 2,135,436,12 and is coordinated by Philips Electronics, The Netherlands. The consortium participants are the Technical University of Eindhoven (Netherlands), the University of Cagliari (Italy), the University of Catania (Italy), the University of Aberdeen (United Kingdom), and the company R2M Solution (Spain).

As an Early Stage Researcher (ESR) and Marie-Curie Fellow in the project **PhilHumans**, my research work focused mostly on *Natural Language Processing (NLP), semantics and sentiment analysis from text in the healthcare domain and beyond.* Besides, being a Marie-Curie Fellow, I was also enrolled as Ph.D. Scholar in the Department of Mathematics and Computer Science of the University of Cagliari. During these years, I worked with *Philips Research* for 18 months and the rest of the 18 months with *UniCa.*

Vivek Kumar
Cagliari, Italy

Contents

List of Tables

List of Figures

Nomenclature

Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
NLP	Natural Language Processing
CML	Classical Machine Learning
TF-IDF	Term Frequency-Inverse Document Frequency
CBOW	Continuous Bag of Words
SVM	Support Vector Machine
kNN	k-Nearest Neighbours
CNN	Convolutional Neural Network
LSTM	Long-Short Term Memory
USE	Universal Sentence Encoder
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pretrained Transformer
LM	Language Model
KG	Knowledge Graph
KGE	Knowledge Graph Embedding
MI	Motivational Interviewing
MITI	Motivational Interviewing Treatment Integrity
MISC	Motivational Interviewing Skill Code

Chapter 1

Introduction

The past decade has seen an explosion in the amount of digital information generated within the healthcare domain. This digital data or information exists in images, video, speech, transcripts, electronic health records, clinical records, and free text. The information encoded in digital data is of innumerable use to create content-based services to assist patients and medical practitioners [145, 15, 51, 37, 29, 128, 52, 141, 82]. For instance, the knowledge extracted from the data can be used to provide new healthcare services globally, addressing the problems related to people's social or economic status. But, the analysis and interpretation of healthcare data is a daunting task, and it demands a great deal of time, resources, and human effort. Therefore, the need for intelligent tools and reliable systems to tackle the complexity of the healthcare domain has increased multi-fold. It has also drawn the interest of a wide research community from healthcare [67, 32, 79, 66], including mental health and its subdomains such as depression, anxiety or substance abuse [70]. Apart from data privacy issues, domain complexity, rigorous accuracy and reliability standards and data scarcity [57, 25, 61] are the key challenges that hamper the real-world application of clinical NLP. While humans have the innate ability to elicit previously acquired knowledge and conveniently integrate it with newly learned concepts to solve tasks at hand, language models significantly fail to do. Therefore, they are limited in scalability and are very task-specific. This thesis aims to address the challenges mentioned above in the real-world application of NLP. It is a blueprint for building intelligent NLP systems capable of understanding the domain peculiarities and semantics applied to healthcare and beyond.

1.1 Thesis Organization

This thesis cumulatively presents the diverse research work with a unifying goal of developing reliable classification systems for better semantic interpretation in the health-

care domain and beyond. The thesis is presented in two parts. The first part, comprising the following two chapters, focuses on best practices for modeling data, feature engineering, and injecting domain knowledge to enrich data semantics applied to tackle several classification tasks in the healthcare domain and beyond.

The second part, comprising of Chapters 4 and 5, presents the high-quality dataset of expert-annotated MI (AnnoMI), its in-depth analysis, augmentation techniques to generate data, and fairness and bias assessments of the AnnoMI and datasets created through employing augmentation techniques on AnnoMi. Due to the diversity of research questions, representations, and methodologies, the chapters are self-contained, i.e., each chapter includes its own literature, motivation, methodology, experiments, and results analysis. Finally, I conclude the thesis with a summary of contributions and a detailed discussion of future work that considers more practical applications of the proposed methods and how LMs can help in domain adaptation. The chapters are organized as follows:

Part I: NLP Practices and Domain Adaptation for Identifying Morbidity from Electronic Health Records

Chapter 2 provides fine-grained discussion on the impact of feature selection, feature representation, stopwords, and different strategies of data preprocessing in classification tasks. Our propositions are validated after conducting a large number of experiments and experimental results in the healthcare domain.

- **Challenges:** Unbalanced and limited training data size are major problems that prevent the classification models from optimal and reliable classification.
- **Contribution:** I have used CML and DL approaches with five pre-trained word embedding and four bag-of-words representations coupled with different feature selection algorithms to identify morbidity conditions within clinical notes. The experimental results prove that single classifiers obtain unstable performances in the presence of small datasets. In contrast, ensemble approaches mitigate this instability and, simultaneously, increase the accuracy of the overall classification.

Chapter 3 introduces a language model (K-LM) to inject domain knowledge directly in the form of triples to solve diverse NLP downstream tasks.

- **Challenges:** The conventional method uses knowledge graph embedding to infuse domain knowledge which at times is not able to capture the semantics.
- **Contribution:** A LM for using world knowledge in the form of triples to solve domain adaptation problems.

Part II: Generating Motivational Interviewing Dataset and its Benchmarking Evalutaion

Chapter 4 introduces AnnoMI - An expert annotated dataset of mental health domain.

- **Challenges:** Mental health is considered a complex domain in healthcare for the real world application of NLP. Due to the lack of publicly accessible data on mental health, it becomes even more difficult to access the reliability of NLP approaches.
- **Contribution:** The first ever public dataset in MI consists of high and low-quality counseling therapy and its in-depth analysis.

Chapter 5 explains the augmentation techniques used along with fairness and bias assessment of AnnoMI.

- **Challenges:** Insufficient training data is a major problem that prevents CML and DL approaches from reliable performance.
- **Contribution:** Provide heuristics methods to augment the AnnoMI dataset and analysis of newly created datasets post augmenting AnnoMI.

1.2 Publications

The work in this thesis primarily relates to the following peer-reviewed articles.

Journal

1. Zixiu Wu, Balloccu S, **V. Kumar**, Rim Helaoui, Reiter E., Diego Reforgiato Recupero and Daniele Riboni, "Creation, Analysis and Applications of AnnoMI, a Dataset of Expert-Annotated Counselling Dialogues". Future Internet 2023, 15, 110. https://doi.org/10.3390/fi15030110.
2. **V. Kumar**, D. R. Recupero, R. Helaoui and D. Riboni, "K-LM: Knowledge Augmenting in Language Models within Scholarly Domain" in IEEE Access 2022. DOI: 10.1109/ACCESS.2022.3201542.
3. **V. Kumar**, D. R. Recupero, D. Riboni and R. Helaoui, "Ensembling Classical Machine Learning and Deep Learning Approaches for Morbidity Identification from Clinical Notes," in IEEE Access 2020. DOI: 10.1109/ACCESS.2020.3043221.

Conference/Workshop Publications

4. **Kumar, Vivek, et al.** "How do you feel? Information Retrieval in Psychotherapy and Fair Ranking Assessment" - Accepted in European Conference of Information Retrieval (ECIR), Dublin, Ireland -2023. (Presented and Under Final Publication)

5. **Kumar, V.**; Balloccu, S.; Wu, Z.; Reiter, E.; Helaoui, R.; Recupero, D. and Riboni, D. (2023). Data Augmentation for Reliability and Fairness in Counselling Quality Classification. In Proceedings of the 1st Workshop on Scarce Data in Artificial Intelligence for Healthcare - SDAIH, ISBN 978-989-758-629-3, SciTePress, pages 23-28. DOI: 10.5220/0011531400003523

6. Zixiu Wu, Balloccu S, **Vivek Kumar**, Rim Helaoui, Reiter E., Diego Reforgiato Recupero and Daniele Riboni (2022). "Anno-MI: A Dataset of Epert Annotated Counselling Dialouges." In The International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2022, Singapore.

7. Zixiu Wu, Rim Helaoui, **Vivek Kumar**, Diego Reforgiato Recupero and Daniele Riboni (2020). "Towards Detecting Need for Empathetic Response in Motivational Interviewing." In SAMIH'20 Workshop of International Conference on Multimodal Interaction (pp. 497-502) 2022 ACM ICMI-MLMI.

8. Dessì, D., Helaoui, R., **Kumar, V.**, Reforgiato Recupero, D., and Riboni, D. (2020). "TF-IDF vs word embedding for morbidity identification in clinical notes: An initial study." In 1st Workshop on Smart Personal Health Interfaces, ACM IUI, SmartPhil 2020 (Vol. 2596, pp. 1-12) CEUR-WS.

Part I : NLP Practices and Domain Adaptation for Identifying Morbidity from Electronic Health Records

Chapter 2

Leveraging NLP approaches for better semantic understanding of text in healthcare domain

2.1 Introduction

In the last years, we have observed a rise in life expectancy, which has also increased the risk of long-term diseases such as diabetes, cognitive impairment, and many other severe health issues [130, 95, 9, 119]. A further downside of a longer lifespan is that people can be affected by more than one disease at a time, leading to the likelihood of under-standard quality of life. An individual with long-term diabetes, for example, has a higher risk of hypertension, high cholesterol levels, blockage of the arteries or veins. According to the World Health Organization report [82], 40% of the population is exposed to at least one long-term health condition, and 25% of the population suffers from multimorbidity in a developed country. According to [82], given that 25% of the world population is already suffering from multimorbidity, its early identification is paramount for preventing the severe health issues which can happen in the future to patients. Therefore, this work aims to automatically identify the multimorbidity factors indicated in the patient's clinical records. Morbidity identification is of great significance in assisting healthcare personnel with several downstream tasks involving handling large volumes of electronic health records. For the experiments, a dataset is used that contains the clinical records of patients, indicating the presence of one or more morbidity factors. In addition, DL models and advanced word embedding representations have recently proven to be state-of-the-art for many NLP tasks and are popularly used within many healthcare problems. Hence, in order to exploit their advantages, the representation of clinical records by methods such as word embedding and bag-of-words

in combination with feature selection techniques using CML and DL approaches are used. The work focuses on discovering whether patients suffer from single or multiple morbidity conditions by studying their past clinical records.

2.2 Related Work

This section briefly reviews the existing artificial intelligence (AI), and NLP methods within the healthcare domain and shows the contribution of the feature selection techniques and word embedding representation.

2.2.1 Artificial Intelligence in Healthcare

The use of DL techniques to identify multimorbidity in clinical reports have been extensively studied in recent years. For instance, DL models in [153] are fed by word and entity embedding to the following two layers, Convolutional Neural Network (CNN) and second Max Pooling. The model improved the results that are obtained during the *i2b2*[1] obesity challenge in 2008. Another work [108] proposed DL based approaches for morbidity status identification. It is focused on automatic learning from the clinical records and feature discovery to disengage hand-crafted feature selection using single and multi-channel CNN models. The single-channel CNN model used an embedding layer to train the model, whereas the multi-channel model employed multiple CNN models in parallel, as an ensemble of CNN models, where each used different hyper-parameters. One more work [77] investigated the performances of long-short term memory (LSTM) networks for entity recognition based on character and word-level representations. The proposed LSTM model outperformed traditional state-of-the-art methods, such as the conditional random field for entity recognition. Authors in [135] uncovered the implementation of sentiment analysis techniques for patient discharge summaries classification. The proposed hybrid model used a semi-supervised technique based on the vector space model and statistical methods in conjunction with an extreme learning machine auto-encoder. The goal is to examine and evaluate the treatment quality based on the discharge summaries. The work presented in [138] investigated the DL approaches, which used pre-trained language models on relation extraction from clinical records. The authors applied pre-trained and fine-tuned BERT, showing that the fine-tuned method performed better than the feature-based method.

[1] https://www.i2b2.org/NLP/Obesity/

2.2.2 Word embedding Models

Clinical records are mostly in the form of free text, which is unstructured, contains typographical errors, and is comprised of healthcare domain-specific terminologies [71]. The representation of these clinical records in a way that they can be used effectively by CML and DL approaches remains one of the top challenges within the healthcare domain. The work in [63] provides a guide for training word embedding on clinical text data. It discusses the different types of word representations, clinical text corpora, available pre-trained clinical word vector embeddings, intrinsic and extrinsic evaluation, applications, and limitations of these approaches. Authors in [151] leveraged the infused elementary distance matrix to update the topic distributions for calculating the corresponding optimal transports. This strategy provides the update of word embedding with robust guidance, improving the algorithmic convergence. As an initial study, the paper [33] presented a comparative analysis of CML and DL approaches with different types of feature representations, such as Term Frequency-Inverse Document Frequency (TF-IDF) and word embeddings.

2.2.3 Feature Selection

Feature engineering in NLP involved creating specific numerical functions to represent salient aspects of the text it requires significant domain knowledge and efforts to identify meaningful features. Feature selection is extensively used to reduce data by eliminating irrelevant and superfluous attributes from the dataset [127, 53]. This technique enhances the data interpretation, improves data visualization, reduces the training time of learning algorithms, and improves prediction performances [58]. The work in [74] mentions the effectiveness of feature selection algorithms in several applications and highlights the challenges faced due to the unique characteristics of data. In work performed in [56], the authors aimed to achieve an affordable, fast, and objective diagnosis of the genetic variant of oligodendroglioma by combining the feature selection with ensemble-based classification. In addition, the work in [46] presented a method called *FREGEX*, which is based on regular expressions to extract features from biomedical, and clinical notes. It is used as a substitute for the n-grams-based feature selection method and employed the algorithms Smith-Waterman and Needleman-Wunsch for sequence alignment. The three datasets used to evaluate the proposed method's performances are manually annotated and contained information on smoking habits, obesity, and obesity types. The features extracted by *FREGEX* based on regular expressions improved the performance of SVM and Naive Bayes based classifiers. The work in [133] used a modified differential evolution algorithm to perform feature selection for cardiovascular disease and optimization of selected features. It also evaluated several performance

measures for the prediction of heart disease to combine the modified differential evolution algorithm with a feed-forward neural network and fuzzy analytical hierarchy process.

2.3 Problem Formulation, Dataset, and Preprocessing

This section provides the formulation of the problem addressed, the used dataset, and the related preprocessing steps applied for CML and DL models.

2.3.1 Problem Formulation

This work aims at a multi-label classification problem to identify morbidity conditions from patients' clinical records. In literature, several approaches exist to tackle the multi-label classification problem [31]. A straightforward and widely used one is to decompose the multi-label problem into multiple binary classification tasks known as *binary relevance method* in the literature [109]. Another approach is to transform the multi-label problem into a single-label multi-class classification problem in which the classes are all label combinations. Since I address the recognition of 16 morbidities in this work, the number of possible classes (i.e., co-morbidities) would be $2^{16} = 65,536$. Therefore, this approach is ruled out, as the number of classes would be too large with respect to the size of the training set. Other more complex solutions exist, including using a multi-label ensemble classifier built from a committee of (single-label) multi-class classifiers or customized machine learning (ML) algorithms adapted to the multi-label problem. Since this work's primary goal is to comprehensively compare different ML approaches and feature extraction techniques, I have adopted a broad and straightforward classification strategy, i.e., the binary relevance method in which the multi-label classification task is decomposed into sixteen binary classification problems.

2.3.2 Dataset Description

The research study is performed on the *n2c2*[2] dataset released for the *i2b2* obesity and co-morbidity detection challenge in 2008. The dataset is completely anonymized by replacing the personal and sensitive information of patients with surrogates. The dataset contains clinical records of patients, and these records indicate that patients may have one or more morbidity conditions from a range of sixteen morbidity conditions (diseases). The sixteen morbidity conditions are *Asthma*, *CAD*, *CHF*, *Depression*, *Diabetes*, *Gallstones*, *GERD*, *Gout*, *Hypercholesterolemia*, *Hypertension*, *Hypertriglyceridemia*, *OA*,

[2] https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/

Obesity, *OSA*, *PVD*, and *Venous Insufficiency*. Originally the *n2c2* dataset contains six documents, out of which Training Textual Judgments, Training Intuitive Judgments, Test Textual judgments, and Test Intuitive Judgments are annotated. The remaining two documents, namely Training Obesity Patients Records and Test Obesity Patients Records, contain the clinical records and a unique *id* associated with them. The textual judgment documents contain all sixteen morbidity conditions, and within each morbidity condition, there is a specific number of ids and labels associated with them. The labels in textual judgment documents can obtain values in {Y, N, U, Q}, where "Y" means yes, the patient has the morbidity, "N" means no, the patient does not have the morbidity, "U" means the morbidity is not mentioned in the record, and "Q" stands for questionable whether the patient has the morbidity. Besides, intuitive judgment documents represent clinical records where domain experts (doctors) are able to infer if those are indicative of having one or more morbidity conditions for the underlying patients. Hence, possible intuitive judgments are limited to labels "Y," "N," and "Q" because "U" is irrelevant as an intuitive judgment. The length of the clinical records is in the range of 500 to 1200 words. A sample of each of the six annotated documents of the morbidity condition *Asthma* is shown in Table 2.1.

2.3.3 Data Preprocessing

The *n2c2* dataset used for experiments contains abbreviations, some typos, punctuation, stopwords, etc., so some preprocessing steps are thus necessary. In this work, I have used two types of feature representations, namely bag-of-words and word embeddings. For bag-of-words, I have employed TF-IDF, whose vector representation relies on the word's occurrence frequency. On the other hand, the word embeddings' working principle is based on capturing the semantic relationships among words. The works in [68, 36] discuss the process and impact of document preprocessing in NLP tasks. Accordingly, the preprocessing steps are performed for transforming the input dataset to be used with the bag-of-words models and are reported below:

- Lower-casing the text to represent the same words of different cases such as *Asthma* and *asthma* as one, i.e., *asthma*.
- Tokenization of text to build a function f, where for each word w, the function f is associated with an integer index i.
- Punctuation and numeric values removal from the text.
- Lemmatization of the tokens.

Table 2.1: Sample data of *n2c2* for class Asthma

Training Documents	Test Documents
Training Data-Textual Judgments <diseases source="intuitive"> <disease name="Asthma"> <doc id="1" judgment="U"/> <doc id="2" judgment="Y"/> <doc id="10" judgment="U"/>	**Test Data-Textual Judgments** <diseases source="intuitive"> <disease name="Asthma"> <doc id="3" judgment="Y"/> <doc id="5" judgment="U"/> <doc id="8" judgment="U"/>
Training Data-Intuitive Judgments <diseases source="intuitive"> <disease name="Asthma"> <doc id="1" judgment="N"/> <doc id="4" judgment="N"/> <doc id="10" judgment="Q"/>	Test Data-Intuitive Judgments <diseases source="intuitive"> <disease name="Asthma"> <doc id="3" judgment="Y"/> <doc id="5" judgment="N"/> <doc id="9" judgment="Y"/>
Training-Obesity Patients Records <doc id="1"> <text> 490646815 | WMC | 31530471 | | 9629480 | 11/23/2006 12:00:00 AM | ANEMIA | Signed | DIS | Admission Date: 11/23/2006 Report Status: Signed\break\ Discharge Date: 6/20/2006\break\ ATTENDING: TRUKA, DEON XAVIER M.D. SERVICE: BH .anList Medical Center. PRIMARY DIAGNOSIS: Congestive heart failure…	Test-Obesity Patients Records} <doc id="3"> <text> 470971328 | AECH | 09071283 | | 6159055 | 5/26/2006 12:00:00 AM | PNUEMONIA | Signed | DIS | Admission Date: 4/22/2006 Report Status: Signed Discharge Date: 7/27/2006 ATTENDING: CARINE , WALTER MD SERVICE: PRINCIPAL DIAGNOSIS: Anemia and GI bleed….

- TF-IDF matrix generation from input data to transform each clinical note into feature vectors.

In order to study the impact of stopword removal for the experiments with word embedding representation, I have preprocessed the input data to generate two sets of feature vectors. One set of feature vectors contains the stopwords, while the other set does not. In the second case, stopwords removal has been performed by using the NLTK[3] library. Furthermore, these two feature vectors are separately used to train the CML models to observe the impact of stopwords on the classifier's performance.

Transforming input data for training of DL models

The dataset must be in integer encoded format to employ DL models and word embedding representation, where a unique integer represents each word. Therefore, to model the data for DL models, the input data is also padded to have symmetrical length throughout, in addition to integer encoding as mentioned below:

- Encoding the input texts into numeric integer representations using vocabulary-index relation. For instance, consider the sentence s: *the patient is asthmatic*, and a function f that maps *the* to "5", *patient* to "34", *is* to "10" and *asthmatic* to "87". Then, the resulting integer-encoded sentence $s_{encoded}$ will be [5, 34, 10, 87].

- Padding each of the input text (integer encoded) to a length equivalent to (average + standard deviation) number of tokens. Most clinical texts are around the average length for the dataset, and the remaining few clinical texts are too long. In this work, I have computed the padding length equal to the sum between the average and the standard deviation of the number of tokens each input text had. This formula has been found empirically on the data and turned out to be a good trade-off between the size of the padding and the length of the document. For example, for four clinical records with 25, 39, 44, and 80 tokens, respectively, the average length is avg=47, and the standard deviation is $std = 20.29$. Hence, the length that is considered for padding is 67.

2.4 Features Representations

This work used bag-of-words TF-IDF and word embedding representations to generate feature vectors. On the one hand, TF-IDF has served as a baseline for many NLP tasks [156] for decades and has proven to be very useful. On the other hand, word embedding is the current state-of-the-art due to their innate capability of capturing the

[3] https://www.nltk.org/

semantics and contextual information for textual features representation of words and text sequences [90, 64].

2.4.1 Term Frequency and Inverted Document Frequency

To generate the feature vectors using bag-of-words TF-IDF representation, I have used the TF-IDF Vectorizer[4] from the scikit-learn library. The experiments are performed with four types of feature vectors using the TF-IDF representations: **All Features** (where feature selection is not applied) and the ones obtained by applying three feature selection algorithms: **ExtraTreesClassifier**, **InfoGainAttributeEval**, and **SelectKBest**. The reason for limiting the number of features is to reduce the computational time for training the models by keeping only those features that contribute most in distinguishing the instances of the different classes.

- **ExtraTreesClassifier** is essentially an ensemble learning method that conceptually shares a similar working principle as that of Random Forest (RF). The only difference is the method for constructing decision trees. For a given set of m features, which are selected randomly from the features set of the input data, ExtraTreesClassifier[5] selects the top features based on their importance (it can be typically calculated by the Gini Index [22]). These random samples of features are further used to create mutually correlated decision trees. This process helps to minimize the chances of overfitting and ranks the features in descending order.

- **InfoGainAttributeEval** is used for feature selection based upon measuring how each feature contributes to decreasing the overall entropy [50]. Entropy is basically a measure of the impurity degree in the dataset. The data is characterized as less impure when the entropy is closer to zero. Hence, the usefulness of an attribute is identified by its contribution to reducing the overall entropy. It can be represented by:

$$InfoGain(Class, Attribute) = H(Class) - H(Class|Attribute)$$

  Where H is the information entropy.

- **SelectKBest** takes the score function as a parameter, which is applied to a pair (m, y) where m corresponds to the features of the input data and y to the corresponding labels. The score function returns an array of scores, one for each feature $m[:, i]$ of m. SelectKBest[6] then simply retains the first k features of m

[4] https://tinyurl.com/y8jqmscd
[5] https://tinyurl.com/ybnzo8rh
[6] https://tinyurl.com/y5c7w6bo

with the highest scores.

The parameter $vocabulary$ of the TF-IDF vectorizer should be provided with a custom list of words (vocabulary) to use the feature selection algorithms from the Python library. This custom vocabulary contains the words (features) in ranked order provided by feature selection algorithms based on the features' information gain. The configuration is set to $max\_features$=600 and $vocabulary$=$custom\_vocab$, where $custom\_vocab$ is the vocabulary of ranked features selected by applying the feature selection algorithms. This setting generates the feature vectors matrix of $\{n \times 600\}$ dimension, where n is the number of text documents (clinical notes).

2.4.2 Word Embeddings

This section describes the general working principle of the word embedding, followed by the details of all the word embedding used for the experiments: pre-trained word2vec, domain-trained, GloVe, fastText, and USE embeddings. They are reported below. Word embedding are distributed representations that model words' properties into vectors of real numbers in a predefined vector space, capturing features and preserving their semantic relationships. As an outcome of this representation, the words having similar meanings have a similar representation. Figure 2.1, presents the visualization of 300-dimensional word embedding of 18586 words generated from *n2c2* dataset using the word2vec model in high dimensional space using Tensorboard[7]. From the visualization, one can note how the words are mapped near to those whose word embedding have a similar meaning. For instance, in the case of the word *diabetes*, the words *diabetic* and *insulinotherapy* are represented in the close semantic space, notable by their scores 0.772 and 0.777.

- **Pre-trained Word2Vec**: *Word2Vec* is an algorithm invented by Google for training word embedding that relies on the distributional hypothesis [85]. The distributional hypothesis uses skip-gram or Continuous Bag of Words (CBOW) algorithms. In the CBOW model, for a given context, the objective is to predict the focal word. The CBOW model with a softmax loss function is essentially a log-linear classification model. The aim is to determine the most likely parameters of the embedding vectors, which can be represented by Equation 2.1:

$$P(w_f|w_c) = \frac{exp(w_f^T w_c)}{\sum_{i=1}^{V} exp(w_f^T w_c)} \tag{2.1}$$

[7] https://projector.tensorflow.org/

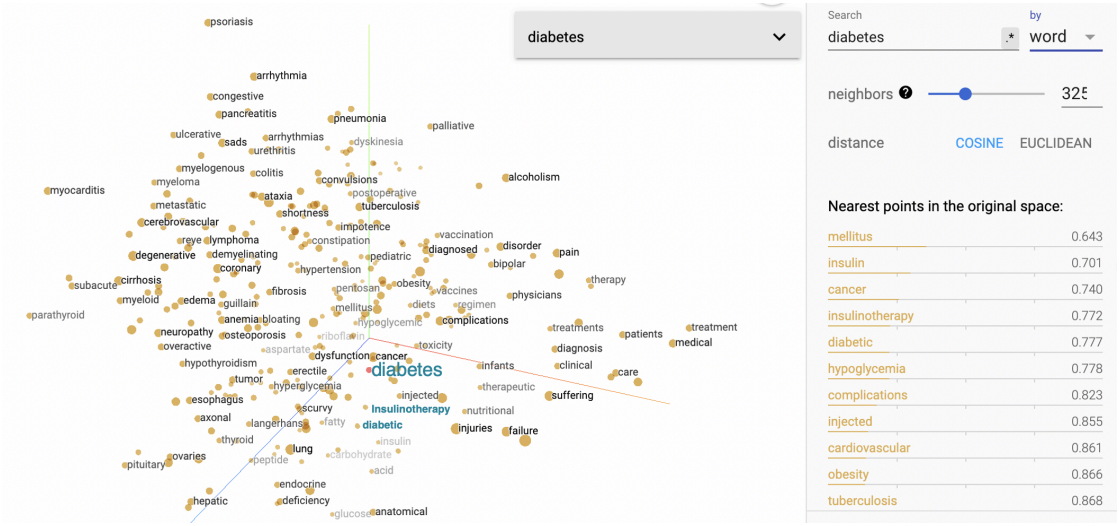


Figure 2.1: Visualizing the semantic relationships between words by Word2Vec word embedding representation.

where w_c is the context (one or more words), w_f is the focal word, and V is the vocabulary size. On the other hand, the skip-gram model can be considered as a complementary model to the CBOW model in terms that its objective involves predicting a context word given a single focal word [63]. The skip-gram model is represented by Equation 2.2:

$$P(w_f|w_c) = \sum_{c=1}^{C} \frac{exp(w_f^T w_c)}{\sum_{i=1}^{V} exp(w_f^T w_c)} \tag{2.2}$$

The *Word2Vec* algorithm aims to detect the meaning and semantic relations by studying the co-occurrences among words in a given corpus. In this work the pre-trained Word2Vec[8] model is used, which is trained on the part of the Google News dataset (about 100 billion words). This pre-trained model contains vectors of three million words and phrases, which are represented in 300-dimensional space.

- **Domain-trained Word2Vec**: The domain-trained word embedding are generated by using the *Word2Vec* algorithm on the $n2c2$ dataset. The rationale for using these embeddings is their advantage in representing the out-of-vocabulary words due to training on the target domain (in our case, healthcare). For this work, word embedding of 300 dimensions with 10 epochs and a window size of 5 by using the Gensim[9] library is generated.

[8] https://code.google.com/archive/p/word2vec/
[9] https://radimrehurek.com/gensim/

- **GloVe**: generator algorithm was developed as an open-source project at Stanford in 2014 [98]. For a given context, to identify how frequently the words appear, *GloVe* utilizes a statistics-based matrix to compute the vectors' scores based on the co-existence of words within the context. Unlike the *Word2Vec* algorithm, *GloVe* uses both the skip-gram model, which is a local context window and the latent semantic analysis method, which belongs to the global matrix factorization methods. For this work, the pre-trained *GloVe6B*[10] embedding model, trained by the Stanford NLP Group on 600 billion tokens of Wikipedia[11] and Gigaword[12] with dimension 300 is used.

- **fastText**: One drawback of *Word2Vec* and *GloVe* algorithms is the fact that they are not able to handle out-of-vocabulary words. To overcome this limitation, Facebook proposed *fastText*[13], which is essentially an extension of the *Word2Vec* algorithm [62, 10, 84]. *FastText* extends the *Word2Vec* skip-gram model by considering internal sub-word information. Basically, words are represented as n-gram of characters instead of learning vectors for words directly. For instance, for n=3, the word *apple* consists of *app*, *ppl*, and *ple*. *FastText* does not consider the internal structure of the word and represents a bag-of-words model with a sliding window over a word. Also, as long as the characters are contained in the window, it is unaffected by the order of the n-grams. This approach helps the model to compute word representations of out-of-vocabulary words and allows the model to understand suffixes and prefixes because it is very likely that some of the n-grams also appear in other words.

- **Universal Sentence Encoder** (USE): While the common practice with word embedding focuses on representing the word, the technique to represent the sentence through a single vector is unclear. To address this, Google introduced pre-trained embedding models known as *USE*, which are optimized to train with a longer text sequence than a single word such as phrases, sentences, and short paragraphs [21, 28]. The pre-trained USE[14] model is trained on several domains with a variety of data sources to accommodate a wide variety of natural language understanding tasks dynamically. It transforms the text into high-dimensional vectors by performing an encoding. It comes with two variations, i.e., one trained with a transformer encoder and the other trained with the deep averaging network. For this work, I have used the deep averaging network pre-trained USE, which takes

[10] https://nlp.stanford.edu/projects/GloVe/
[11] https://dumps.wikimedia.org/enwiki/
[12] https://catalog.ldc.upenn.edu/LDC2011T07
[13] https://fastText.cc/docs/en/english-vectors.html
[14] https://tfhub.dev/google/universal-sentence-encoder/4

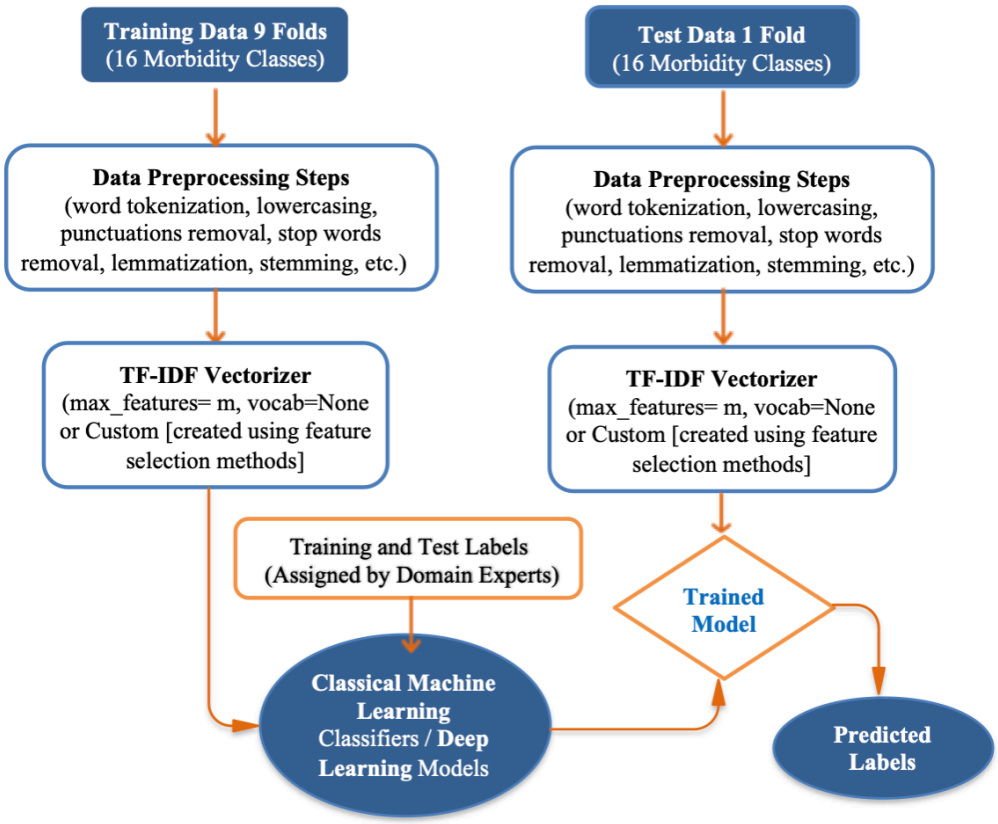


Figure 2.2: The architecture of the pipeline for morbidity detection in clinical records using TF-IDF representations with CML and DL approaches.

variable-length English texts as input and outputs 512-dimensional vectors.

2.5 Classification Models

This work has employed two types of classification models based on CML and DL approaches with each type of feature representation mentioned in Section 2.4. Figure 2.2 shows the generalized architecture of the pipeline used for the classification of clinical records using TF-IDF representations with CML and DL approaches. The pipeline consists of training and testing phases. Prior to the training stage, the preprocessing is applied to the clinical records, as mentioned in Section 2.3.3. After that, classifiers are trained on the feature vectors derived from the training samples. After creating feature vectors, the previously trained classifiers predict each clinical record label in the testing sample. Finally, the performances of different classifiers are evaluated by calculating standard metrics such as precision, recall, and F-1 score. The CML and DL models and their architectures are mentioned below.

2.5.1 Classical Machine Learning Models

Experimental results reported in this paper are obtained using standard implementations of CML algorithms provided by the Weka toolkit using Python Weka-Wrapper[15] interface with Java Virtual Machine[16] environment. I have employed Support Vector Machine (SVM) [30], k-Nearest Neighbours (kNN) [1], Naive Bayes [60], Random Forest [14], Random Tree [106], J-48 [118] and J-Rip [27].

2.5.2 Deep Learning Models

The DL models have used two types of representations, one with word embedding and the other with bag-of-words.

- **Deep Learning Models Used with Word Embeddings:** The DL model used in this work for word embedding representations is the network with an embedding layer, two Bidirectional Long Short-Term Memory (BiLSTM) layers, a dense layer followed by an output layer for the binary classification task. Figure 2.3 presents the related architecture. The embedding layer is initialized by the following four inputs:

  - *input_dim* (size of the vocabulary);
  - *output_dim*: (dimension of the dense embeddings);
  - *weights (embeddings_matrix),* and
  - *input_length* (length of input sequences).

  The *input_dim* represents the length (V) of the unique vocabulary created from the input data (clinical records). The input matrix (integer encoded vectors) has dimension $\{n \times m\}$, with n equal to the number of clinical records and *input_length* corresponding to m, which is the maximum number of tokens considered for each text. The *embeddings_matrix* is the vector representation of the corresponding words of the vocabulary and has dimension $\{V \times x\}$, where x represents the *output_dim*. Specifically, *output_dim* for all the embedding is 300 except USE, which has a value of 512. The output of the embedding layer is passed to two hidden layers that implement BiLSTM neural networks [78]. LSTM is a particular kind of recurrent neural network that can store the history of the input data and has already proven to be able to find patterns in data where the sequence of the information matters [24]. By using the bidirectional version, the models can

[15] https://pypi.org/project/python-weka-wrapper/
[16] https://pypi.org/project/javabridge/

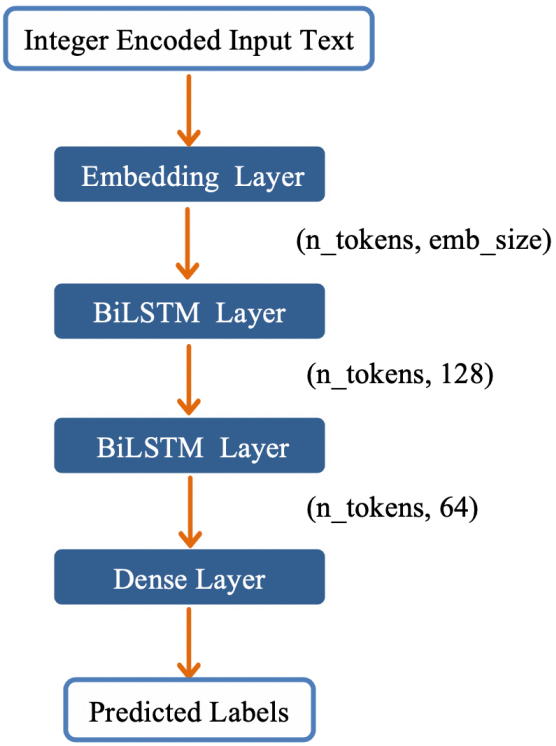


Figure 2.3: The architecture of DL models to use word embedding representation

learn from the input data both backward and forward. Finally, the output of the BiLSTM layer is fed to a fully connected dense layer to predict the labels.

- **Deep Learning Models used with Bag-of-words representation:** For the bag-of-words model TF-IDF representation is used, in conjunction with the employed feature selection algorithms. The differences between this and the above-mentioned DL model are the following:

  - This model does not have an embedding layer and the input is directly fed to the BiLSTM layer.
  - Secondly, in this model the input data do not undergo the preprocessing steps such as integer encoding and padding when used with TF-IDF representation.

  The input to the BiLSTM layer, in this case, is the TF-IDF matrix, which is generated by the TF-IDF vectorizer and has dimension $\{n \times 600\}$, with n the number of text documents (clinical records). Figure 2.4 presents the architecture of the DL network used with TF-IDF representation.

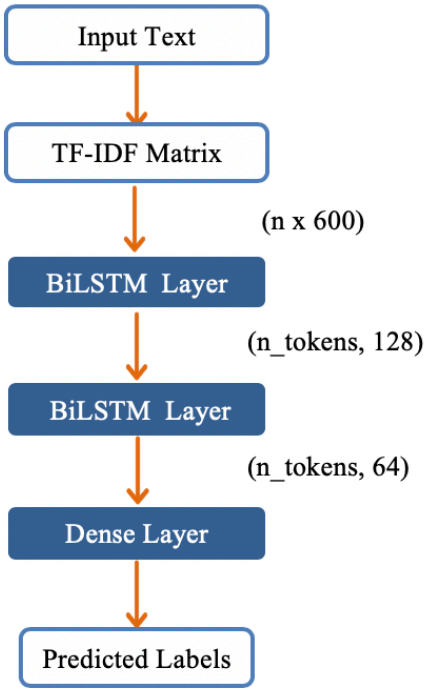


Figure 2.4: The architecture of DL models to use TF-IDF representation.

2.6 Experiments and Results

The specifications of computational resources to run the experiments are summarized in Table 2.2.

Table 2.2: Server Specifications.

Item	Specification
CPU	Intel Core i3-7100 (-HT-MCP-) CPU @ 3.90 GHz
GPU	NVIDIA GP102 [TITAN X], 12 GB memory
Graphic driver	NVIDIA graphic driver version 440.33.01
CUDA	Version 10.2
OS	Ubuntu (17.10)
Python	Version 3.6.6

The experiments are performed with CML and DL approaches using the bag-of-words applied to feature selection algorithms and word embedding representations. Ensemble learning is also employed over a large number of combinations of classifiers to improve the single model performances and obtain stable results. To ensure the ro-

bustness of performance estimation and avoid the bias of the single ML models, 10-fold cross-validation is used [65]. The performances of different classifiers and feature representations are measured in terms of F-1 score (F-1) using micro and macro averaging over 10 folds provided by the scikit-learn[17] library. The formulas to calculate accuracy, precision, recall, and F-1 score are given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

where TP, FP, and FN represent each label's true positive, false positive, and false negative, respectively. The experiments that have been carried out can be divided into three groups for ease of understanding, which are mentioned below:

1. In the first set of experiments, CML and DL approaches are used with bag-of-words representations coupled with feature selection algorithms using TF-IDF representation, as mentioned in Section 2.4.1.

2. In the second set of experiments, CML and DL approaches are used with word embedding generated by pre-trained models of word2vec, domain-trained with word2vec, GloVe, fastText, and USE embeddings. The feature vectors generated by these pre-trained word embedding to train CML classifiers are generated from the same input data by either keeping or removing the stopwords. The purpose of generating two sets of feature vectors is to study the relatedness of stopwords with the context of the text and their impact on the classifier's performance. The DL models are trained only with the feature vectors of the input data with stopwords as the standard experiment.

3. As the last set of experiments, the ensemble learning approach is implemented on a large number of combinations of classifiers to improve the single model performances.

The following subsections describe the three sets of experiments.

[17] https://tinyurl.com/y4mt646z

2.6.1 Experimental Results with bag-of-words coupled with feature selection algorithms

This subsection provides the details of the experiments performed with CML and DL approaches with bag-of-words coupled with feature selection algorithms using TF-IDF representations. TF-IDF evaluates the importance of a feature based on its frequency. Identifying features that contribute the most to distinguish the classes is useful for improving the models' performances. Thus this work has adopted three feature selection algorithms, namely ExtraTreesClassifier, InfoGainAttributeEval, and SelectKBest, along with the **All Features**. Table 2.3 depicts the results of CML classifiers with **All Features** using TF-IDF representations. Tables 2.4, 2.5, and 2.6 illustrate the results of CML Classifiers with feature selection algorithms ExtraTreesClassifier, SelectKBest, and InfoGain, respectively. Finally, Table 2.7 includes the results of the DL models with the four bag-of-words applied to feature selection algorithms using TF-IDF representations. The key observations from the performed experiments are listed below:

- In general, the feature selection algorithms have improved the performance of CML classifiers (typically by 1%). The two best-performing classifiers with **All Features** are SVM and RF with 98.45 and 98.1 micro F-1 scores, respectively (as shown in Table 2.3). Using the ExtraTreesClassifier as the feature selection algorithm has improved the micro F-1 score of RF to 98.82 and SVM to 99.26 (shown in Table 2.4), which is the best performance of CML classifiers among all the experiments.

- In contrast, the Naive Bayes classifier used with **All Features** has the best performance with a Micro F-1 score of 89.31 (as shown in Table 2.3) than with any feature selection algorithms.

- In the case of DL approaches, **All Features** using TF-IDF has been outperformed by the feature selection algorithms achieving up to 13% of F-1 score (shown in Table 2.7).

- The reason for the low performance of DL models with **All Features** using TF-IDF is because that TF-IDF selects the features based on the frequency of the words, which is not useful to distinguish the morbidity classes. Feature selection algorithms identify the most important features that allow the DL models to learn clinical records' context, further improving classification performances.

- From the experimental results, it turned out that using feature selection algorithms has shown more benefit on DL models than on CML algorithms. In fact,

with **All Features**, the micro F-1 score of DL models is 76.47, whereas, with the usage of ExtraTreesClassifier, it has improved to 89.63 (as shown in Table 2.7).

- As far as the computational time and resource requirements are concerned, the CML models have proven to be computationally faster and less demanding. The training time of the CML models seen so far is up to 600 seconds while of the DL models it is much higher (a couple of hours) using the same machine mentioned in Table 2.2 (DL approaches employed both the CPU and the GPU whereas the CML models just the CPU).

Table 2.3: Performances of CML Classifiers with All Features using TF-IDF Representations.

	J-48		**J-Rip**		**Naive Bayes**		**Random Forest**		**Random Tree**		**SVM**		**KNN (k=1)**		**KNN (k=5)**	
Morbidity Class	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	**99.4**	**98.75**	98.4	97	91.1	81.6	**99.4**	**98.75**	98.9	97.95	98.7	97.55	98.1	96.45	89.6	76.65
CAD	93.6	93.4	94.3	94.05	88.4	87.95	97.7	97.55	96.4	96.2	**99.2**	**99.1**	96	95.8	70.8	67.45
CHF	96	96.05	94	94.05	86.9	86.95	97.8	97.75	96.2	96.2	**98.9**	**98.9**	94.7	94.7	68	64.45
Depression	95.4	93.3	95.1	93.15	84.7	76.85	96.3	94.45	95.2	93.2	**97**	**95.75**	96.7	95.2	79.5	59.85
Diabetes	95.7	94.85	96	95.25	89.3	86.9	96.3	95.45	96.5	95.8	**96.9**	**96.25**	93.5	92.6	73.2	71.75
Gallstones	99.1	98.45	99.4	98.95	89.4	82.05	98.8	97.85	98.4	97.15	**99**	**98.2**	98.8	97.85	83.4	51.75
Gerd	97.3	96.15	96.4	94.95	88.1	83.25	98.1	97.25	97.4	96.35	**97.9**	**96.95**	97.2	96.05	82.5	68.7
Gout	**99.7**	**99.3**	**99.7**	**99.3**	95.2	89.2	99.2	98.1	98.4	96.45	99.6	99.1	98.7	97.15	90.2	68.2
Hyperchol-esterolemia	97.4	97.35	91.7	91.55	82.9	82.55	97.5	97.4	95.7	95.6	**97.7**	**97.7**	95.7	95.6	78.9	78.2
Hypertension	**97.8**	**96.25**	95.5	92.75	86.4	76.9	97.7	95.95	97.5	95.7	97.6	96.1	95.2	92.5	85.1	72.1
Hypertrigl-yceridemia	98.2	89.95	98.8	94.1	96.7	86.6	**99.4**	**96.9**	**99.4**	**96.9**	**99.4**	**96.9**	99.2	95.95	94.5	48.6
OA	97.8	96.8	97.1	95.75	88.7	83.2	97.6	96.35	97	95.55	**98.5**	**97.75**	96.1	94.4	75.1	65.2
Obesity	**99**	**99**	97.6	97.6	80.9	80.75	96.8	96.75	96.1	96.1	97.2	97.15	96.3	96.25	76	74.85
OSA	99.5	98.95	**99.8**	**99.6**	89.2	76.55	98.6	96.95	97.8	95.35	98.8	97.4	98.5	96.8	89.4	72.65
PVD	98.1	96.25	98.7	97.45	92.1	82.05	98.5	96.9	98.3	96.65	98.9	97.85	**99.4**	**98.7**	90.1	73.65
Venous Insufficiency	97.3	90.1	97.4	90.85	99	96.3	**100**	**100**	99.5	98.3	**100**	**100**	98.8	95.95	94.5	68.05
Average	97.58	95.93	96.86	95.39	89.31	83.72	98.1	97.14	97.41	96.21	**98.45**	**97.66**	97.05	95.74	82.55	67.63

Table 2.4: Performances of CML Classifiers with Feature Selection Algorithm ExtraTreesClassifier using TF-IDF Representations.

	J-48		**J-Rip**		**Naive Bayes**		**Random Forest**		**Random Tree**		**SVM**		**KNN (k=1)**		**KNN (k=5)**	
Morbidity Class	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	99.1	98.15	99.1	98.15	73.7	67	99.2	98.35	98.3	96.75	99.2	98.35	**99.4**	**98.75**	86.2	54.55
CAD	95.5	95.35	94.7	94.45	93	92.7	98.5	98.45	97	96.9	**99.6**	**99.55**	97	96.9	83.4	81.05
CHF	96	96	97.2	97.25	86	85.95	98.9	98.9	97.3	97.35	**99.1**	**99.1**	98.2	98.25	86	85.8
Depression	94.7	92.45	96.5	95.1	75.1	72.75	96.9	95.4	96.5	94.95	**99.2**	**98.85**	96.3	94.45	78.1	48.85
Diabetes	96.7	96.1	95.6	94.75	93.8	92.75	**98**	**97.55**	96.9	96.35	97.8	97.3	96.7	96	79.2	67.5
Gallstones	97.6	95.75	99.1	98.45	70.7	66.4	99	98.2	99.4	98.95	**99.8**	**99.65**	99	98.2	83.4	50.35
Gerd	97.3	96.1	95.8	94.15	76	73.6	98.8	98.35	97.4	96.4	**99.8**	**99.65**	98.4	97.65	78.2	48.2
Gout	99.4	98.65	99.8	99.55	85.3	76.9	99.8	99.55	99.2	98.2	**100**	**100**	99.2	98.1	87.6	48.25
Hypercholesterolemia	95.3	95.25	91.7	91.6	87.8	87.45	**98.4**	**98.35**	97.5	97.45	98.2	98.15	97.9	97.95	82.3	81.95
Hypertension	97.5	95.7	96	93.4	71.4	67.1	97.9	96.35	97	95.15	**99.2**	**98.6**	97.2	95.25	82.6	52.45
Hypertriglyceridemia	99.2	96.1	98.7	94	91.4	75.25	**99.6**	**98**	99.6	98	99.4	96.9	99.4	96.9	94.5	48.6
OA	97.5	96.35	96.7	95.2	75.4	72.45	98.9	98.4	96.8	95.25	**99.1**	**98.75**	98.5	97.75	80.3	52.6
Obesity	98.2	98.15	97.6	97.6	87.7	87.75	98.9	98.9	96.1	96.1	**99.4**	**99.35**	95.9	95.85	79.6	78.3
OSA	99.3	98.55	**99.8**	**99.6**	79.1	71.25	99.4	98.7	98.6	97.1	99.4	98.7	98.6	96.95	86.8	52.5
PVD	98.1	96.2	97	94.2	77.1	70.1	**98.9**	**97.85**	98.5	97	**98.9**	**97.85**	98.5	96.9	85.7	50.25
Venous Insufficiency	98.1	92.7	97.8	92.15	86.5	71.5	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	92.8	48.15
Average	97.47	96.10	97.07	95.60	81.88	76.93	98.82	98.21	97.88	96.99	**99.26**	**98.80**	98.14	97.24	84.17	59.33

Table 2.5: Performances of CML Classifiers with Feature Selection Algorithm SelectKBest using TF-IDF Representations.

	J-48		**J-Rip**		**Naive Bayes**		**Random Forest**		**Random Tree**		**SVM**		**KNN (k=1)**		**KNN (k=5)**	
Morbidity Class	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	**99.4**	**98.75**	98.8	97.8	91.5	82.55	**99.4**	**98.75**	98.9	97.95	98.9	97.95	97.9	96	89	75.05
CAD	94.8	94.6	91.2	90.9	88.7	88.25	97.2	97.1	94.9	94.65	**99.2**	**99.1**	95.8	95.55	69.2	65.35
CHF	96.2	96.25	95.7	95.7	86.9	86.95	98.2	98.25	96	96.05	**99.1**	**99.1**	95.4	95.35	69.2	65.9
Depression	95.9	94.05	96.1	94.6	84.7	77.2	96.3	94.45	95.5	93.5	**97.1**	**95.75**	95.5	93.55	79.8	60.7
Diabetes	96.1	95.35	95.2	94.35	89.7	87.4	96.5	95.7	94.3	93.2	**97.1**	**96.55**	92.4	91.4	70.1	69.2
Gallstones	**99**	**98.25**	**99**	**98.25**	89.3	81.9	98.8	97.85	97.6	95.85	**99**	98.2	98.8	97.85	83.3	53.05
Gerd	97	95.65	96.6	95.3	88.2	81.25	**98.1**	**97.25**	96.7	95.4	97.7	96.6	97.4	96.35	83.4	70.4
Gout	99.6	99.1	**99.7**	**99.3**	94.9	88.45	99.2	98.1	98.8	97.25	99.6	99.1	98.8	97.35	89.8	70.05
Hypercholesterolemia	97	97	90.6	90.5	83	82.65	97.5	97.45	95	94.85	**97.7**	**97.7**	94.7	94.7	78.9	78.55
Hypertension	97.5	95.75	93.9	90.4	86.5	77.05	**97.7**	95.95	95.8	93	**97.7**	**96.1**	96.6	94.5	86.8	72.85
Hypertriglyceridemia	98.4	91.4	98.8	94.4	96.6	86.3	99.4	96.9	99.4	96.9	99.4	96.9	**99.6**	**98**	94.5	48.6
OA	97.2	95.95	97	95.7	88.2	82.55	97.6	96.35	97	95.55	**98.3**	**97.4**	96.4	94.75	77.8	67.95
Obesity	**98.8**	**98.8**	97.7	97.75	80	79.85	96.3	96.25	95.3	95.2	97	96.95	96.1	96.05	76.7	76.35
OSA	99.5	98.95	**99.8**	**99.6**	89.3	76.7	98.6	96.95	97.8	95.5	98.8	97.4	98.2	96.3	90.7	78.65
PVD	98	96	97.4	94.95	91.8	81.55	98.5	96.9	98.5	97.05	98.9	97.85	**99**	**98.05**	90	73.2
Venous Insufficiency	97.6	90.65	97.3	90.4	98.5	94.85	**100**	**100**	99.8	99.15	**100**	**100**	98.6	95.2	95.2	74.05
Average	97.63	96.03	96.55	94.99	89.24	83.47	98.08	97.14	96.96	95.69	**98.47**	**97.67**	96.95	95.68	82.78	68.74

Table 2.6: Performances of CML Classifiers with Feature Selection Algorithm InfoGain using TF-IDF Representations.

	J-48		**J-Rip**		**Naive Bayes**		**Random Forest**		**Random Tree**		**SVM**		**KNN (k=1)**		**KNN (k=5)**	
Morbidity Class	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	**99.4**	**98.75**	98.8	97.8	91.5	82.55	**99.4**	**98.75**	98.9	97.95	98.9	97.95	97.9	96	89	75.05
CAD	94.8	94.6	91.2	90.9	88.7	88.25	97.2	97.1	94.9	94.65	**99.2**	**99.1**	95.8	95.55	69.2	65.35
CHF	96.2	96.25	95.7	95.7	86.9	86.95	98.2	98.25	96	96.05	**99.1**	**99.1**	95.4	95.35	69.2	65.9
Depression	95.5	93.45	96.1	94.6	84.7	77.2	96.3	94.45	95.5	93.5	**97.1**	**95.75**	95.5	93.55	79.8	60.7
Diabetes	95.9	95.15	95.2	94.35	89.7	87.4	96.5	95.7	94.3	93.2	**97.1**	**96.55**	92.4	91.4	70.1	69.2
Gallstones	97.4	95.25	99	98.25	89.3	81.9	98.8	97.85	97.6	95.85	**99**	**98.2**	98.8	97.85	83.3	53.05
Gerd	96.3	94.6	96.6	95.3	88.2	83.35	**98.1**	**97.25**	96.7	95.4	97.7	96.6	97.4	96.35	83.4	70.4
Gout	99.6	99.1	**99.7**	**99.3**	94.9	88.45	99.2	98.1	98.8	97.25	99.6	99.1	98.8	97.35	89.8	70.05
Hypercholesterolemia	97	97	90.6	90.5	83	82.65	97.9	97.9	95	94.85	**99.6**	**99.1**	94.7	94.7	78.9	78.55
Hypertension	97.2	95.35	93.9	90.4	86.5	77.05	97.5	95.65	95.8	93	**97.7**	**96.1**	96.6	94.5	86.8	72.85
Hypertriglyceridemia	97.7	87.45	98.8	94.4	86.5	77.05	99.4	96.9	99.4	96.9	99.4	96.9	**99.6**	**98**	94.5	48.6
OA	97.2	95.95	97	95.7	88.2	82.55	97.6	96.35	97	95.55	**98.3**	**97.4**	96.4	94.75	77.8	67.95
Obesity	**98**	**97.95**	97.7	97.75	80	79.85	96.3	96.25	95.3	95.2	97	96.95	96.1	96.05	76.7	76.35
OSA	99.1	98.1	**99.8**	**99.6**	89.3	76.7	98.6	96.95	97.8	95.5	98.8	97.4	98.2	96.3	90.7	78.65
PVD	97.5	95.1	97.4	94.95	91.8	81.55	98.5	96.9	98.5	97.05	98.9	97.85	**99**	**98.05**	90	73.2
Venous Insufficiency	97.6	90.65	97.3	90.4	98.5	94.85	**100**	**100**	99.8	99.15	**100**	**100**	98.6	95.2	95.2	74.05
Average	97.28	95.29	96.55	94.99	88.61	83.02	98.10	97.15	96.96	95.69	**98.59**	**97.75**	96.95	95.68	82.78	68.74

Table 2.7: Performances of DL models with Bag-of-Words Coupled with Feature Selection Algorithms using TF-IDF Representations.

	Extra Tress Classifier		**InfoGain**		**SelectKBest**		**All Features**	
Morbidity Class	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	90.86	80.25	**92.22**	**82.14**	84.86	45.89	84.87	45.90
CAD	**83.02**	**82.33**	67.85	54.05	60.51	38.35	60.09	43.16
CHF	**86.3**	**85.91**	74.01	69.64	52.98	40.37	54.20	53.38
Depression	**86.15**	**76.46**	81.41	66.7	76.86	43.43	76.86	43.45
Diabetes	**86.93**	**83.54**	80.2	72.35	70.2	41.21	70.20	41.24
Gallstones	**96.08**	**90.66**	85.24	58.12	82.52	45.2	82.52	45.21
Gerd	**90.9**	**86.26**	80.18	58.74	77.15	43.5	77.15	43.55
Gout	95.61	**86.4**	**95.91**	86.08	87.45	46.62	87.45	46.65
Hypercholesterolemia	**78.64**	**76.12**	67.07	60	56.61	35.96	56.84	51.45
Hypertension	**81.31**	**49.54**	79.2	47.86	**81.31**	44.53	**81.31**	44.84
Hypertriglyceridemia	**97.13**	**83.71**	94.56	50.02	94.46	48.56	94.46	48.57
OA	**90.35**	**81.51**	85.22	75.04	78.37	43.91	78.37	43.93
Obesity	**94.83**	**94.69**	72.36	66.57	55.48	35.53	55.48	35.68
OSA	**97.88**	**95.08**	94.66	88.32	85.91	46.2	85.91	46.21
PVD	**85.72**	**67.9**	85.08	64.62	85.07	45.96	85.07	45.96
Venous Insufficiency	92.3	**53.39**	**92.77**	48.08	**92.77**	48.08	**92.77**	48.12
Average	**89.63**	**79.61**	83.00	65.52	76.41	43.33	76.47	45.45

2.6.2 Experimental Results With Word Embeddings

In this group of experiments, the CML and DL approaches are trained with the embedding generated by the pre-trained word2vec and domain-trained with word2vec, fastText, GloVe, and USE models. The results of the experiments are summarized in Tables 2.8, 2.9, 2.10. In particular, Tables 2.8 and 2.9 present the results of CML classifiers using the word embedding representation with the input data without the removal of stopwords (raw) and with the input data not containing the stopwords (pre-processed), respectively. The best performances of CML classifiers with word embedding representations extracted from Tables 2.8 and 2.9 are shown in Figure 2.7. Moreover, for ease of understanding, Figure 2.8 represents the performance of the CML and DL classifiers with bag-of-words coupled with feature selection algorithms. Figure 2.9 shows the CML and DL classifiers' plots with word embedding representation. The winning configurations are highlighted for each kind of used representation.

The key observations from the performed experiments are listed below:

- The CML classifiers have performed only slightly better (less than 1% of difference) with embedding when the input data do not contain the stopwords. The case when the input data contain the stop words has lower performances, where the domain-trained and USE embedding are the exceptions.

- Given the small size of the used dataset and the minimal difference between the two kinds of CML models (with and without stopwords), any concrete conclusion can not be made for their performance based on the presence of stopwords in the dataset. However, it can be assumed that, given the technical terminology used within the clinical notes, stopwords should not play an important role while preprocessing the dataset. A more detailed analysis of them is out of the scope of this work and will be investigated in a future direction.

- In the case of DL models, the use of word embedding has further improved their performance with respect to the bag-of-words representation coupled with feature selection algorithms. The best performance of the DL model is observed when GloVe word embedding are employed with 94.3 average micro F-1 scores (Table 2.10) against the average micro F-1 score of 89.63 when used with bag-of-words representation (Table 2.7). Besides, the former corresponds to the best performance of DL models for all sets of experiments.

- Generally, it is expected that the domain-specific word embedding will perform better (due to the absence of out-of-vocabulary words) than pre-trained word embeddings, but it does not happen if the training data is small. The small amount

Table 2.8: Performances (averaged over all the morbidity classes) of CML classifiers with Word embedding when input data contain stopwords.

	Domain-Train		**fastText**		**GloVe**		**Word2Vec**		**USE**	
Classifier	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
J-48	92.21	87.79	93.34	89.65	92.95	88.84	92.62	86.39	**93.42**	**89.8**
J-Rip	86.12	77.87	**90.16**	**85.49**	89.16	83.7	88.76	80.92	89.56	83.83
Naive Bayes	58.14	51.5	65.89	**63.83**	63.51	60.84	61.96	60.47	**68.13**	60.33
Random Forest	97.92	96.95	98.03	96	97.98	95.9	96.63	92.85	**98.06**	**97.1**
Random Tree	97.19	95.69	**97.5**	**96.33**	97.1	95.68	95.87	92.88	96.98	95.52
SVM	79	51.2	89.08	78.07	86.13	70.78	87.31	71.85	**90.06**	**85.31**
KNN (k=1)	97.18	95.74	**97.51**	**96.07**	97.18	95.82	95.95	93.04	97.31	95.95
KNN (k=5)	81.79	65.52	**84.12**	68.14	83.38	66.76	83.13	65.71	83.16	**68.4**

of data, in fact, jeopardize the chances of learning the subtle peculiarities of the domain and will lead to the high variance estimation of the model's performance. For such a reason, the performances of the DL models with domain-trained embedding are worse than those of the other four pre-trained embeddings. In contrast to the DL models, the performance of CML models using domain-specific word embedding is only slightly affected by the small size of the dataset.

- Regarding the computational time, the CML models have again turned out to be computationally fast and less resource exhaustive as compared to the DL models. The training time of the CML models ranges between 80 and 600 seconds. The reason for the reduced training time with respect to the CML models employing bag-of-words is the lower dimension of the embedding vectors (typically 300-dimensional for all types of word embeddings except USE, which has 512).

- Different from the CML classifiers, the training time of the DL classifiers has increased up to 40 hours. The reason for this higher computational cost lies within the employment of new layers of deep neural networks.

The comparison of the training time between the CML and DL models is presented in Figures 2.5 and 2.6. Finally, Table 2.10 presents the results of DL models with the word embedding representation.

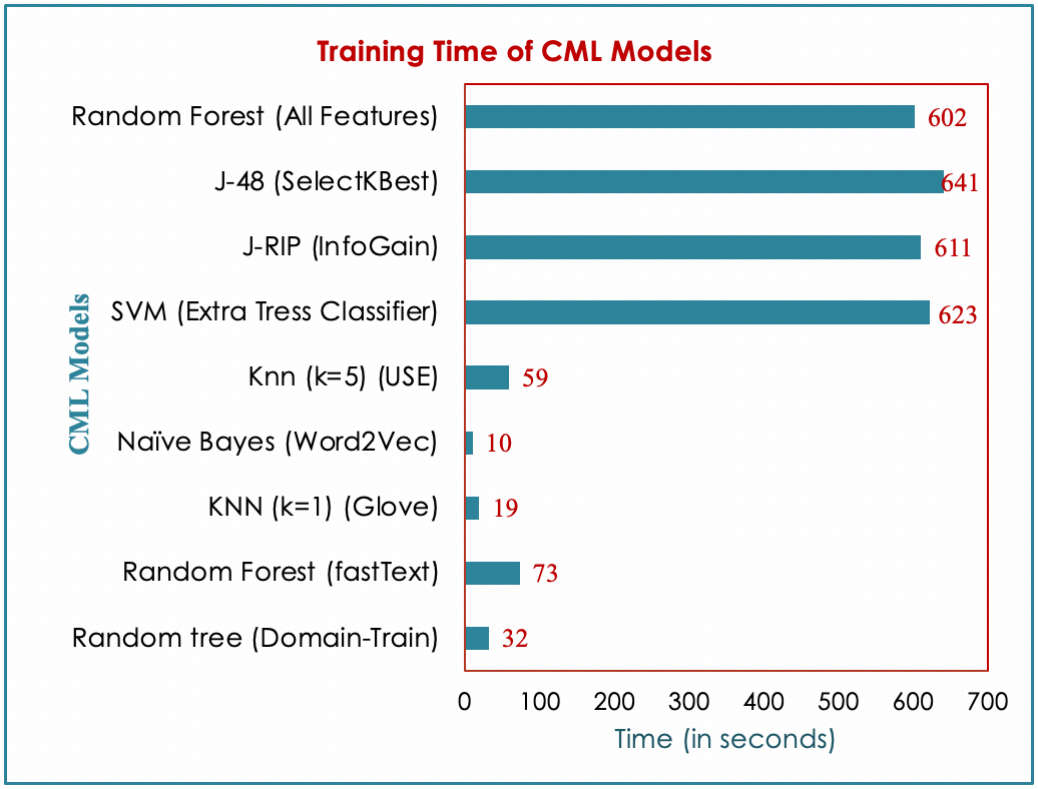


Figure 2.5: The training time of CML models with different representations.

Table 2.9: Performances (averaged over all the morbidity classes) of CML classifiers with Word embedding when input data do not contain stopwords

Classifier	Domain-Train		fastText		GloVe		Word2Vec		USE	
	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
J-48	91.79	87.09	93.44	89.59	**93.65**	89.85	93.36	**89.91**	92.53	87.91
J-Rip	85.09	75.21	89.78	84.43	89.86	**85.53**	**89.91**	84.62	88.34	81.77
Naive Bayes	57.5	52.25	**69.25**	**66.96**	66.5	64.48	65.71	63.93	51.55	47.02
Random Forest	97.8	96.83	**98.13**	**97.24**	98.05	95.93	98.12	96.08	97.9	96.09
Random Tree	97.25	95.84	97.2	95.74	96.99	95.64	**97.41**	96.07	97.01	**96.61**
SVM	79.43	51.65	**90.27**	81.32	89.49	79.17	89.83	79.91	89.9	**84.42**
KNN (k=1)	97.13	95.54	**97.54**	**96.3**	97.22	96.26	97.43	96.02	97.43	96.19
KNN (k=5)	81.67	64.27	**84.57**	69.12	84.3	68.44	84	68	84.05	**69.69**

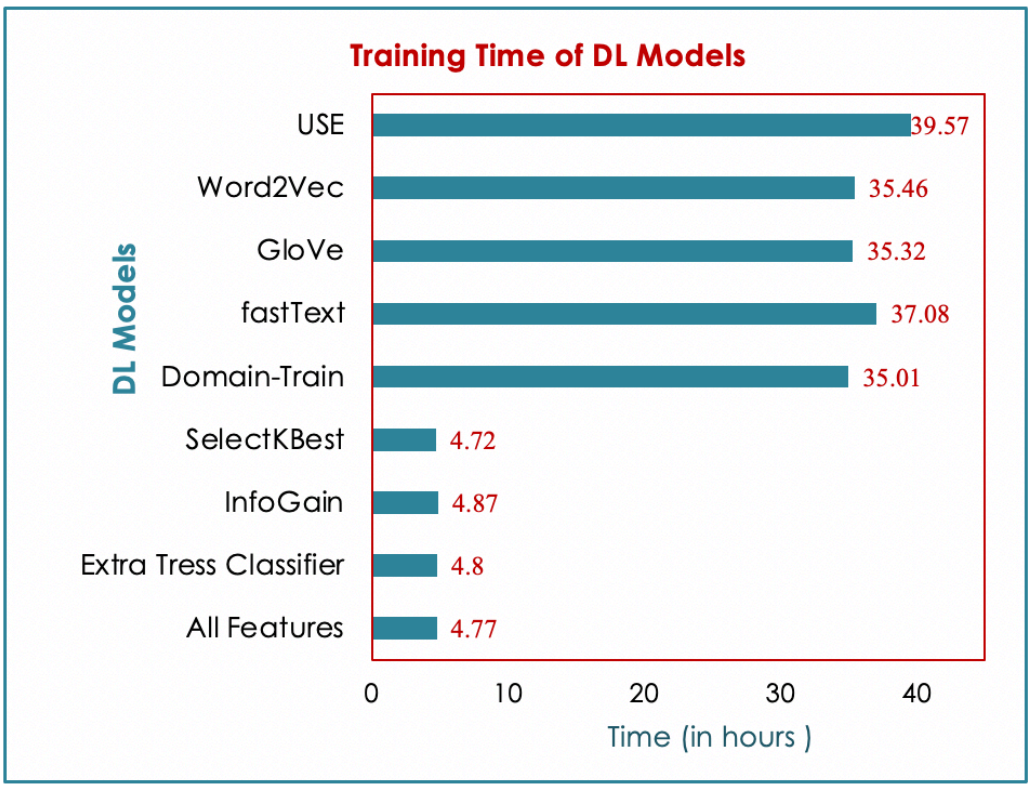
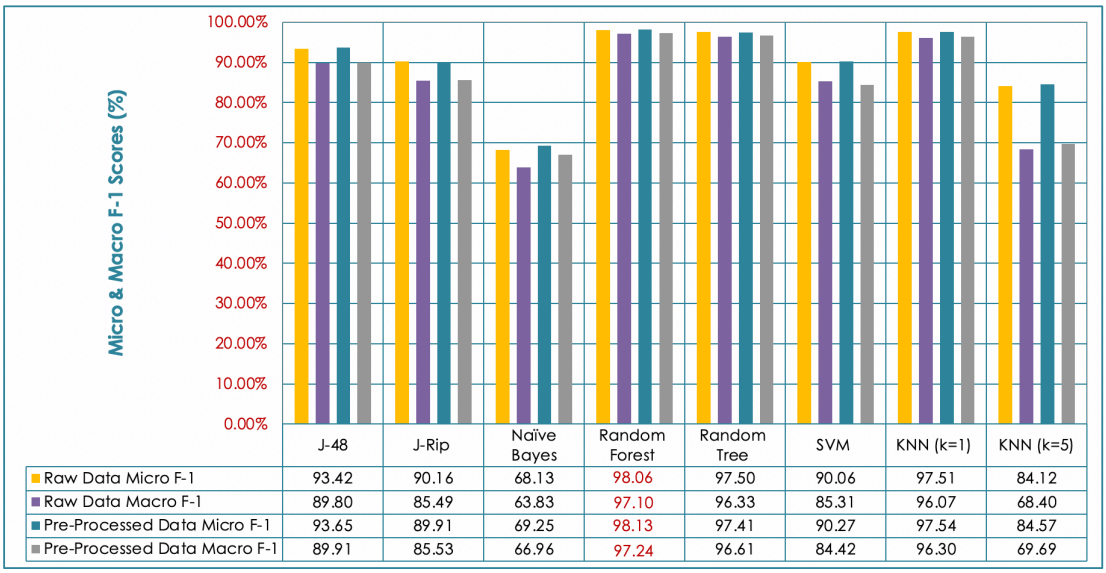


Figure 2.6: The training time of DL models with different representations.



	J-48	J-Rip	Naïve Bayes	Random Forest	Random Tree	SVM	KNN (k=1)	KNN (k=5)
Raw Data Micro F-1	93.42	90.16	68.13	98.06	97.50	90.06	97.51	84.12
Raw Data Macro F-1	89.80	85.49	63.83	97.10	96.33	85.31	96.07	68.40
Pre-Processed Data Micro F-1	93.65	89.91	69.25	98.13	97.41	90.27	97.54	84.57
Pre-Processed Data Macro F-1	89.91	85.53	66.96	97.24	96.61	84.42	96.30	69.69

Figure 2.7: Best performances of CML classifiers using embedding with and without stopwords taken from Tables 2.8 and 2.9.

Table 2.10: Performances of DL models with Word Embeddings.

	Domain-Train		**fastText**		**GloVe**		**Word2Vec**		**USE**	
Morbidity Class	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	86.75	57.51	95.68	90.56	**96.63**	**93.45**	92.85	84.57	87.81	58.77
CAD	60.5	54.4	87.68	87.32	**91.07**	**90.5**	74.08	73.05	80.88	82.73
CHF	61.17	56.74	87.6	87.52	**93.91**	**93.85**	89.36	89.25	81.51	78.35
Depression	80.57	60.54	94.62	91.75	**96.58**	**94.83**	91.73	87.6	88.13	71.26
Diabetes	73.26	56.9	92.15	91.41	**94.8**	**91.73**	91.32	89.72	88.93	85.1
Gallstones	83.53	54	88.32	83.56	**92.46**	**85.96**	89.74	75.65	89.42	69.86
Gerd	78.43	54.49	83.1	74.77	**89.04**	**82.17**	75.74	63.59	86.95	68.82
Gout	88.25	53.96	96.21	90.67	**96.81**	**91.43**	87.96	69.63	91.33	64.02
Hypercholesterolemia	67.56	66.05	88.83	88.12	**91.08**	**90.56**	88.45	88.02	82.42	83.84
Hypertension	79.94	57.29	**97.24**	95.16	97.23	**95.92**	89.91	82.8	89.28	93.54
Hypertriglyceridemia	94.35	61.77	93.36	86.47	98.87	93.92	97.13	72.57	**98.56**	**87.13**
OA	76.97	57.75	88.85	83	**93.26**	**89.33**	82.22	70.02	86.4	67.04
Obesity	55.05	48.72	84.3	83.31	**85.8**	**85.49**	67.52	64.19	64.08	62.36
OSA	86.92	55.38	93.25	84.97	**97.58**	**94.77**	91.85	83.92	92.44	71.69
PVD	86.03	55.86	95.41	90.04	**99.14**	**98.25**	92.53	81.55	91.23	67.87
Venous Insufficiency	92.3	47.95	**97.9**	**86.43**	97.66	84.23	94.04	66.16	87.89	32.8
Average	78.22	56.2	91.53	87.19	**94.3**	**91.21**	87.28	77.64	86.46	71.81

2.6.3 Experimental Results With Ensemble Approach

In this final group of experiments, the ensemble learning approach is discussed. Ensemble learning works by first training each single machine learning model and then combining their predictions. The rationale behind ensemble learning is to take the best from a given set of algorithms by combining their outputs. Given the large number of classifiers employed in this study, it is not feasible to experiment with all possible combinations of employed machine learning algorithms. For this reason, the most effective DL and CML algorithms are selected for applying the ensemble approach. For the ensemble approach, four bag-of-words models with feature selection and five types of pre-trained word embedding are used with eight CML algorithms and BiLSTM-based DL models. Hence, a total of 9×9 = 81 classification models in total are considered for the ensemble. Considering the formula $2^a - (a + 1)$, with $a \geq 2$ equal to the number of models, for calculating the total number of possible ensembles constituted, would account for a total of $(2^{81} - 82)$ possible combinations. Computing all the possible ensembles resulting from the formula above would be unfeasible. Therefore, the number of models for generating the configurations of ensembles is limited. As per the hypothesis, combining CML and DL classifiers in the same ensemble configuration would increase the model's stability without decreasing accuracy. Hence, for this work, the 6 top-performing CML models and the 5 top-performing DL models from the pool of classifiers are included in the ensemble configurations. The ensemble combinations are generated based on r, where r is an odd number between 3 and 11. The choice of using 11 classifiers corresponded to 1013 different ensemble configurations, which is a reasonable number for this experiment. The classifiers selected for the ensemble configurations are listed below:

1. Random Forest classifier used with SelectKBest feature selection algorithm.
2. SVM classifier used with ExtraTreesClassifier feature selection algorithm.
3. kNN classifier (where k=1) used with ExtraTreesClassifier feature selection algorithm.
4. kNN classifier (where k=1) used with fastText word embedding representation.
5. Random Forest classifier used with USE word embedding representation.
6. Random Forest classifier used with fastText word embedding representation.
7. DL model used with USE word embedding representation.
8. DL model used with GloVe word embedding representation.

9. DL model used with fastText word embedding representation.
10. DL model used with InfoGain feature selection algorithm.
11. DL model used with ExtraTreesClassifier feature selection algorithm.

The performance of all the above-mentioned 1013 ensemble combinations is computed, and the results of the six best-performing combinations among them are summarized in Table 2.11. Out of the top six ensemble models, ensembles 1, 3, and 5 consist of five classification models, while 3 classification models constitute ensembles 2, 4, and 6.

Table 2.11: Performances of Ensemble Approaches.

	Ensemble-1		**Ensemble-2**		**Ensemble-3**		**Ensemble-4**		**Ensemble-5**		**Ensemble-6**	
Morbidity Class	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1	Micro F-1	Macro F-1
Asthma	**99.37**	**98.75**	99.05	98.14	**99.37**	**98.75**	99.16	98.34	**99.37**	**98.75**	**99.37**	**98.75**
CAD	99.58	99.55	99.58	99.56	99.47	99.44	**99.68**	**99.67**	99.58	99.56	99.58	99.55
CHF	**99.78**	**99.78**	99.67	99.67	**99.78**	**99.78**	99.67	99.67	**99.78**	**99.78**	99.34	99.34
Depression	98.86	98.37	98.97	98.52	98.86	98.37	**99.07**	**98.67**	98.86	98.37	98.76	98.22
Diabetes	98.27	97.90	98.27	97.91	98.16	97.77	98.16	97.78	98.16	97.77	**98.37**	**98.02**
Gallstones	99.70	99.47	**99.80**	**99.65**	**99.80**	**99.65**	99.70	99.48	99.70	99.47	99.60	99.30
Gerd	98.95	98.49	**99.18**	**98.83**	99.07	98.66	99.07	98.66	98.95	98.49	98.95	98.49
Gout	99.70	99.31	99.70	99.31	99.70	99.31	**99.80**	**99.54**	99.70	99.31	99.70	99.31
Hypercholesterolemia	**98.52**	**98.49**	98.06	98.02	98.40	98.37	98.29	98.26	98.29	98.25	**98.52**	**98.49**
Hypertension	**99.68**	**99.47**	99.58	99.30	99.47	99.12	99.47	99.12	99.47	99.12	99.58	99.30
Hypertriglyceridemia	99.49	97.44	99.49	97.44	99.49	97.44	**99.59**	**97.97**	99.49	97.44	99.39	96.90
OA	98.93	98.39	98.93	98.40	**99.04**	**98.56**	**99.04**	**98.56**	98.93	98.39	98.72	98.06
Obesity	99.03	99.02	99.25	99.24	99.03	99.02	98.71	98.69	98.92	98.91	**99.46**	**99.46**
OSA	99.50	98.94	**99.60**	**99.16**	99.50	98.94	99.40	98.74	99.50	98.94	99.40	98.73
PVD	99.04	98.06	**99.15**	**98.28**	99.04	98.06	99.04	98.06	99.04	98.06	98.93	97.84
Venous Insufficiency	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Average	**99.27**	**98.84**	**99.27**	**98.84**	99.26	98.83	99.24	98.83	99.23	98.79	99.23	98.73

The structure of the top six ensemble combinations is listed below:

- **Ensemble-1**. The number of constituting classifiers for Ensemble-1 is 5, which are: DL models with (fastText and GloVe) word embeddings, SVM with ExtraTreesClassifier algorithm, Random Forest with SelectKBest algorithm, and kNN(k=1) with fastText word embeddings.

- **Ensemble-2**. The number of constituting classifiers for Ensemble-2 is 3, which are: DL model with GloVe word embeddings, SVM with ExtraTreesClassifier, and kNN(k=1) with fastText word embeddings.

- **Ensemble-3**. The number of constituting classifiers for Ensemble-3 is 5, which are: DL models with (fastText and GloVe) word embeddings, SVM with ExtraTreesClassifier algorithm, and kNN(k=1) with ExtraTreesClassifier algorithm, and kNN(k=1) with fastText word embeddings.

- **Ensemble-4**. The number of constituting classifiers for Ensemble-4 is 3, which are: DL model with fastText word embeddings, SVM with ExtraTreesClassifier, and kNN(k=1) with fastText word embeddings.

- **Ensemble-5**. The number of constituting classifiers for Ensemble-5 is 5, which are: DL models with (fastText and GloVe) word embeddings, SVM with ExtraTreesClassifier algorithm, Random Forest with fastText word embeddings, and kNN(k=1) with fastText word embeddings.

- **Ensemble-6**. The number of constituting classifiers for Ensemble-6 is 3, which are: DL model with GloVe word embeddings, Random Forest with SelectKBest algorithm, and kNN(k=1) with fastText word embeddings.

To get the final predictions of the ensembles, the majority voting technique is used, which is generally used for these kinds of tasks [16]. In this technique, multiple models are used to make predictions for each clinical record, and predictions by each model are considered as a ”vote.” For instance, for a document (a clinical record), if three classifiers have predicted the class of a sample as 1, 0, and 1, then the final predicted label will be 1, as it secures more than half the votes. The experimental results are summarized in Table 2.12 for ease of understanding. The first section of Table 2.12 presents the average performances of the eight CML algorithms with each of the four bag-of-words models coupled with feature selection algorithms and the five-word embeddings. The second section of Table 2.12 presents the average performances of the nine DL models used with each of the two representations, i.e., the four bag-of-words models representations and the five-word embedding representations. Lastly, the third section shows

the average performances of all the ensemble models, which are tested with 3, 5, 7, and 9 constituents.

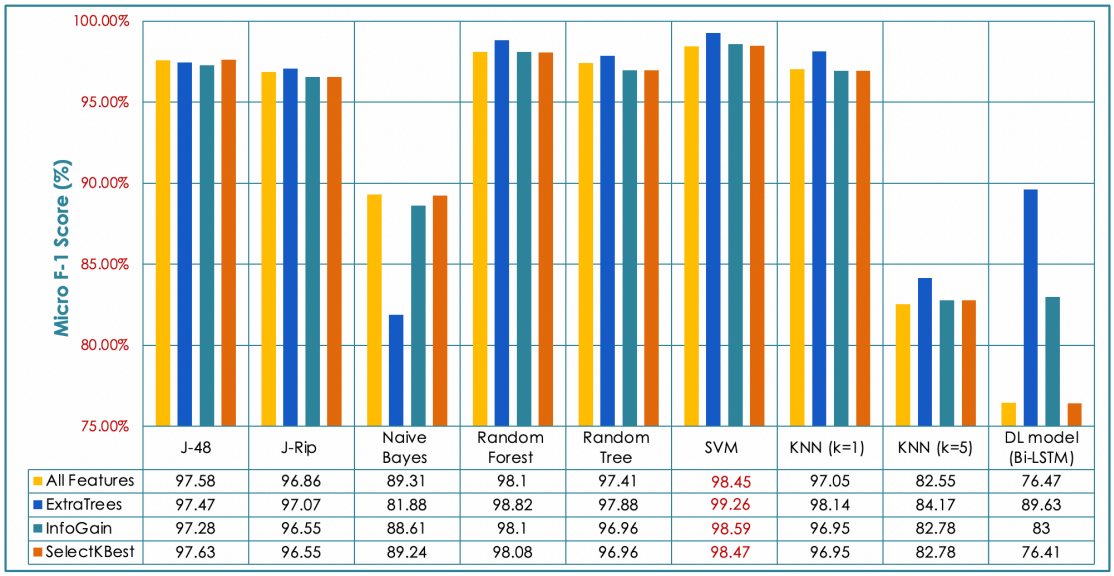
Table 2.12: Average, Best Micro F-1 Score and Standard Deviation of CML and DL Classifiers and Ensembles. Results are averaged over all the morbidity classes.

CML Classifiers Used With	**Average Micro F-1 Score**	**Best Micro F-1 Score**	**Standard Deviation**
All Features	**94.66**	98.45	5.71
Extratrees Algorithm	94.33	**99.26**	7.04
InfoGain Algorithm	94.48	98.59	5.68
SelectKBest Algorithm	94.58	98.47	**5.60**
Domain-Train embedding	85.95	97.25	14.49
Fasttext embedding	90.02	97.54	10.27
Glove embedding	89.53	98.05	11.15
Word2vec embedding	89.47	98.12	11.50
USE Embeddings	87.33	97.43	16.34
DL Approach Used with	**Average Micro F-1 Score**	**Best Micro F-1 Score**	**Standard Deviation**
Bag-of-Words Representations	81.37	89.63	6.31
Word embedding Representations	**87.58**	**94.3**	**6.11**
Ensemble Approach Using	**Average Micro F-1 Score**	**Best Micro F-1 Score**	**Standard Deviation**
3-Models	96.96	**99.27**	2.35
5-Models	97.97	**99.27**	0.96
7-Models	98.34	99.12	0.47
9-Models	**98.51**	98.93	**0.27**
11-Model (just 1)	97.79	97.79	n.a.

The comparison of the aforementioned performances has been done in terms of the average micro F-1 score, best micro F-1 score, and standard deviation. Note that values in each row of the table are averaged over all the morbidity classes and settings within the underlying model. The best performer among them gives an F-1 score of 99.27, with an average of 97.97 and a standard deviation of 0.96. From the results, it is evident that the CML and DL classifiers' performances are lower than the presented ensembles.

2.7 Conclusion and Future Work

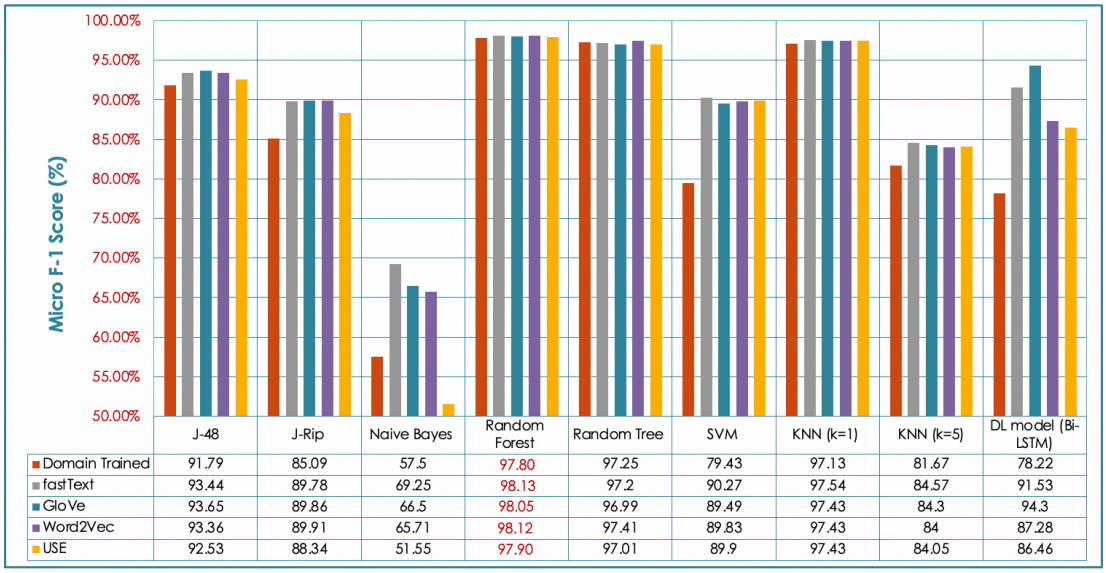
The performance variations of CML and DL classifiers with the different feature vector representations are holistically analyzed. First, the performances of CML classifiers with word embedding with or without stopwords are discussed. The results indicate that the performances of CML classifiers are slightly better when input data do not include the stopwords in general when used with different embeddings, with domain-trained and USE embedding being the exception. Unlike the other word embedding approaches that take a word as input to generate the feature vectors, the input to USE is a sentence. Therefore, the embedding produced by USE for the sentence captures the context of the sentence and the mutual relatedness of words within it. Removing stopwords can change the sentence's meaning, negatively impacting the predictions. In the case of CML classifiers used with bag-of-words representation, the performances of CML classifiers have improved with the ExtraTreesClassifier feature selection algorithm, i.e., SVM with the F-1 score of 99.26, which is the best performance for all the performed experiments. Overall, the CML classifiers have performed better with the feature selection algorithms. Furthermore, in the case of DL approaches, the used feature selection algorithms have substantially improved the model's performance. The F-1 score of 76.25 with **All Features** has increased to 89.63 when ExtraTreesClassifier is used. In the case of the DL approaches used with different word embeddings, GloVe has achieved the best results. In the context of training time, the CML models have proven to be computationally lighter and faster to train. Conversely, the DL models have a long training time, which increases while switching from experiments with bag-of-words to word embedding representations. Finally, the integration of CML and DL approaches by employing the ensemble technique to produce ensemble models has improved the single best classification model's performances. While the best performances of the DL models are achieved with GloVe word embedding obtaining a micro F-1 score of 94.3, the top 989 out of 1013 ensembles got a higher score than it. Although the best ensemble score of 99.27 is only slightly better than the best performance of a single CML model, 99.26, the efficacy of ensemble models can be appreciated by their high average and low standard deviation values. The average micro F1 scores of ensembles made of 3,



	J-48	J-Rip	Naive Bayes	Random Forest	Random Tree	SVM	KNN (k=1)	KNN (k=5)	DL model (Bi-LSTM)
All Features	97.58	96.86	89.31	98.1	97.41	98.45	97.05	82.55	76.47
ExtraTrees	97.47	97.07	81.88	98.82	97.88	99.26	98.14	84.17	89.63
InfoGain	97.28	96.55	88.61	98.1	96.96	98.59	96.95	82.78	83
SelectKBest	97.63	96.55	89.24	98.08	96.96	98.47	96.95	82.78	76.41

Figure 2.8: Experimental results of CML and DL models with and without the employment of feature selection algorithms.

5, 7, and 9 classification models are greater than the average of each single representation technique used for experiments. In addition, while the CML classifiers suffer from a high standard deviation value, the ensembles are much more stable with a standard deviation, which decreases from 2.35 when using 3 classifiers to 0.27 when using 9 classifiers. Despite being computationally intensive, the ensemble method proved to be a viable technique. Indeed, for a highly imbalanced and small dataset like the used in this work, the prediction stability of the model is quintessential. In general, for the minority class, the classification models tend to achieve lower precision or recall scores. Using the ensemble approach, can not only deal better with the prediction of the minority class but also reduce the variance of predictions and, thus, the generalization error. In the context of future work, techniques like data augmentation and state-of-the-art word embedding representations exploiting transformer architecture such as BERT, ELMO, XLNet, etc., can be employed to deal with the constraints of small datasets in order to improve the performances of DL models and the overall ensemble. Moreover, a detailed analysis of the benefits of removing or not the stopwords from the clinical notes will be carried out to understand when they are useful or not in the underlying domain.



	J-48	J-Rip	Naive Bayes	Random Forest	Random Tree	SVM	KNN (k=1)	KNN (k=5)	DL model (Bi-LSTM)
Domain Trained	91.79	85.09	57.5	97.80	97.25	79.43	97.13	81.67	78.22
fastText	93.44	89.78	69.25	98.13	97.2	90.27	97.54	84.57	91.53
GloVe	93.65	89.86	66.5	98.05	96.99	89.49	97.43	84.3	94.3
Word2Vec	93.36	89.91	65.71	98.12	97.41	89.83	97.43	84	87.28
USE	92.53	88.34	51.55	97.90	97.01	89.9	97.43	84.05	86.46

Figure 2.9: Experimental results of CML and DL models with word embeddings.

Chapter 3

Knowledge augmenting practices for domain adaptation using knowledge graphs

3.1 Introduction

Humans have the innate ability to elicit previously acquired knowledge and integrate it with newly learned concepts to solve tasks at hand. However, we have made remarkable progress in developing intelligent systems to mimic human-like abilities in recent times; still, these systems are limited in scalability and are very task-specific. For instance, the transfer learning [96] approach has left a mark in cross-domain adaptation, but it is limited to interrelated domains application. Similarly, the general LM, BERT from Transformers [38], pre-trained on Wikipedia[1] and BookCorpus[2], has given promising results on specific NLP downstream tasks. But in the case of cross-domain adaptability, it lacks task-specific and domain-related knowledge, and hence more detailed fine-tuning strategy analyses are necessary to further improve the performance [73]. To overcome the constraint of domain-adaption, approaches based on augmenting additional knowledge to LMs have turned out to be effective. One way to inject general world knowledge is represented by Knowledge Graphs (KGs), which are well-known interlinked structured knowledge, comprised of triples [42]. A semantic triple, or RDF triple or triple, is the atomic data entity in the RDF data model. As the name indicates, a triple is a set of three items, *subject (s)*, *predicate (p)*, and *object (o)*, that encodes the semantic data (e.g., <Deep Learning, sub-domain, Machine Learning>). The triples are represented as (s, p, o), where the predicate (p) is the relationship between the subject

[1] https://www.wikipedia.org/
[2] https://huggingface.co/datasets/bookcorpus

and the object. A few of the popular general knowledge bases available in the public domain include WordNet[3] [86], Cyc[4] [81], DBpedia[5] [5], YAGO[6] [124], Freebase[7] [11], NELL[8] [20], and Wikidata[9] [134]. The existing methods for injecting the world knowledge into the classification models use KGs in the form of Knowledge Graphs Embedding (KGE). KGEs are low-dimensional representations of knowledge graph's entities and relations while preserving their semantic meaning [136] and are useful for tasks such as KG completion [137], relation extraction [139, 113], entity classification [93, 94], entity resolution [93, 12], etc. In the existing literature, a plethora of knowledge representation approaches such as TransE [13], TransH [137], TransR [75], DistMult [120], ComplEx [129], RotatE [125], HolE [92], ConvE [121], ConvKB [91], DKRL [150] are available to transform triples into KGE. KGE has proven to be useful in incorporating world knowledge, but it is still debatable if the KGEs sufficiently capture KGs semantics [59]. Analyses from this work indicate that leveraging KGEs for semantic interpretability (as in the case of word embeddings) may seem intuitive, but this is not always the case because the performance of KGE is limited and is heavily dependent on the characteristics of the dataset. This conclusion is based on the evaluation of the semantic representation of KGEs. It is observed that for given KG entities, KGEs can learn certain semantic features, but this learning is non-uniform due to the varying quality of semantic representation across different entities within the dataset. These findings raise questions about the applicability and efficacy of KGEs for semantic capturing and link prediction [2, 116, 117] and triple completion. Two of the available existing works tackled the challenge of semantics capturing associated with KGE, by using RDF triples as a direct source of knowledge [76, 3]. However, these works are far from mature and lack to answer some crucial knowledge fronts. For instance, they do not discuss the quantification of knowledge injection for achieving optimal classification performances. They also do not discuss the impact of using general KGs in domain-specific tasks. These limitations motivated our investigation of using triples as the source of semantic knowledge, and the challenges that arise in domain-specific and open-domain NLP downstream tasks. This work proposes an LM (K-LM), which is the combination of BERT [114] and GPT-2 [17, 107] for text classification. K in K-LM stands for knowledge, and LM is the language model. K-LM leverages world knowledge by incorporating additional knowledge in the form of triples to mitigate the prevalent knowledge gap scenarios. The experiments are performed on the *scholarly* domain using KGs of the same

[3] https://wordnet.princeton.edu/
[4] https://cyc.com/
[5] http://downloads.dbpedia.org/wiki-archive/
[6] https://yago-knowledge.org/
[7] https://developers.google.com/freebase/
[8] http://rtw.ml.cmu.edu/rtw/
[9] https://www.wikidata.org/wiki/Wikidata:Main_Page

domain. Furthermore, `Deterministic` and `Non-deterministic` strategies are also introduced to seed the triples in the proposed LM for achieving optimal results in domain-specific and open-domain classification tasks.

3.2 Related Work

This section briefly reviews the existing state-of-the-art techniques for augmenting knowledge in the form of triples for domain-specific and open-domain tasks. The pre-trained BERT model trained on cross-domain text corpora such as BookCorpus[10] and Wikipedia[11] has achieved significant performance on several NLP downstream tasks. Despite this improvement, the transformer model lacks task-specific and domain-related knowledge, which can contribute to further improvements. To address the limitations of domain knowledge gaps in cross-domain knowledge-driven NLP tasks, the work proposed a BERT-based text classification model called BERT4TC [154]. The proposed model focuses on constructing auxiliary sentences and converting the original classification task into a sentence pair with a binary set of new categorical labels. Another work proposed a knowledge-enabled language representation model of BERT (K-BERT) with KGs [76]. The triples are used in the sentences as domain knowledge along with soft-position and a visible matrix to limit the impact of knowledge to tackle the domain. The purpose of the visible matrix is to control the flow of a huge amount of knowledge (also called Knowledge Noise (KN)), as it may lead to a change in the sentence from its actual meaning. Structurally, the English language operates on tokens or word-level embeddings, while the model proposed in this work used character-level embeddings; therefore, this constraint limits its application to NLP tasks in the Chinese language. Furthermore, a severe limitation of the work is that it did not provide any evaluation method to determine the suitability of triples and order them according to the given context. The work performed in [3] proposed an attention mechanism-based DL model that could use knowledge graphs as knowledge support for the task at hand. It can also be considered the first work that attempted to incorporate the KGs in the form of triples directly. It is about a convolution-based model that learns representations of knowledge graph entities and relation clusters by reducing the attention space. The outcome of this work shows that the proposed model is suitable for the classification tasks at hand while being trained on significantly less training data when it has access to world knowledge resources such as KGs. However, the works mentioned above provide methods to use world knowledge, but they significantly lack on several knowledge fronts, such as novel methods to select best-fit triples based on context, quantification of triples in-

[10] https://huggingface.co/datasets/bookcorpus
[11] https://en.wikipedia.org/wiki/Main_Page

jection, and challenges of KN associated with knowledge injection. To bridge these knowledge gaps, my approach introduces a robust pipeline to select and inject triples, starting from the input sentences themselves. Our work also takes into account the "context-aware" and "context-unaware" dependencies of the LM and their impact on the knowledge integration process. I also implement novel approaches that allow to custom feed and rank the input triples to address this challenge. My approaches are comprised of `Non-deterministic` and `Deterministic` methods, which help in quantifying knowledge injection and generalization of direct integration of triples for optimal knowledge injection in the LM. In the context of generalization, I have used two KGs as a source of domain knowledge and performed experiments with incremental integration of triples to draw conclusions about the relevance of the used KGs for the task at hand.

3.3 Problem Formulation, Dataset and Preprocessing

This section presents the problem statement tackled in this work, the details of the used dataset, the preprocessing strategies, and the KGs employed to carry out the experiments.

3.3.1 Problem formulation

The main goal of this work is to infuse domain knowledge in pre-trained language models to equip the models with additional knowledge available in the form of KGs and leverage the added knowledge to achieve higher accuracy in the classification tasks. I have used the K-LM model to tackle a binary classification task in this work. More precisely, for a given text $\boldsymbol{t}$ and a target class $\boldsymbol{c}$, the objective is to infer a function f(t, c) $\longrightarrow$ l that computes 1 if $\boldsymbol{t}$ belongs to the class $\boldsymbol{c}$, 0 otherwise. Here $\boldsymbol{l}$ is the binary label that can only take values in $\{0, 1\}$. To inject the domain knowledge in K-LM, two KGs have been used, namely AI-KG and AI-KG-Small. The details of K-LM and KGs are provided in subsequent sections. This work introduces novel methods of using world knowledge and different processes of integrating the domain knowledge, which is fundamentally different from transfer learning. This work also aims to benchmark the quantity of knowledge injection and introduce novel methods for filtering the triples to inject into classification models.

3.3.2 Dataset Description

The research study is performed on the *Scholarly Domain*. We have named it *Scholarly Domain* for ease of understanding and it is used throughout the paper. *Scholarly*

Domain indicates *Goggle Scholar*[12] that provides a simple way to search for scholarly literature (research papers/articles/books). The experimental dataset is comprised of 30,023 abstracts of research papers in Computer Science from 2001 to 2019. The labels 1 and 0 represent if the underlying research paper belongs to the AI domain or not. The distribution of the dataset is: Label-0 = 9,720, Label-1 = 20,303. From the given distribution of the scholarly dataset, it is evident that it is unbalanced, and the majority class is label 1, representing that most of the papers belong to the AI domain. Standard preprocessing steps such as lowercasing the text, special characters removal, and tokenization [67, 68, 36, 131, 32] are adopted to prepare the textual data. The input dataset is further divided into the train, validation, and test sets. The train set is used to fine-tune the model, while the model's performance evaluation is done on the validation and test sets. The distribution of test, validation and train sets of the dataset is 60%, 20%, and 20%, respectively. For ease of understanding, the notations used throughout the paper are listed in Table 3.1.

Table 3.1: Notations Used

Uni-sub-triple	Triple extracted from AIKG and AI-KG-Small whose subject is a unigram (one word).
Bi-sub-triple	Triple extracted from AIKG and AI-KG-Small whose subject is a bigram (two words).
Tri-sub-triple	Triple extracted from AIKG and AI-KG-Small whose subject is a trigram (three words).
AI-KG	The Artificial Intelligence Knowledge Graph used as the source of world knowledge.
AI-KG-Small	The subset of Artificial Intelligence Knowledge Graph.
Full KG	When all the triples of an entire input KG, are used.
K-LM	The Language Model proposed in the paper which is a combination of GPT-2 and BERT.
N	The total number of triples present in the given KG (AI-KG and AI-KG-Small in our case).
U	The total number of unique triples present in the given KG. Unique triples are defined as the triples having distinct *'subject'* entities.
L	The length of the entity 'subject' of the triples. It is measured by the number of words present in the subject. For $\{l = 1, 2, 3\}$ a given triple is called *Uni-sub-triple*, *Bi-sub-triple* and *Tri-sub-triple*, respectively.
TIT	For a given KG and dataset used for our experiments, the total injectable triple is the number of triples that can be injected into the input sentence while fine-tuning K-LM. A triple is considered *'injectable'* when its subject is present in the input sentence.

[12] https://scholar.google.com/

3.3.3 Knowledge Graphs

For our work, KGs are used as a medium to infuse world knowledge for *Scholarly Domain* and the details of the used KGs are mentioned below.

- **AI-KG**- The Artificial Intelligence Knowledge Graph (AI-KG) is an automatically generated large-scale knowledge graph comprised of 857,658 research entities. AI-KG consists of 1,2 million statements and 14 million triples which are extracted from 333K research publications belonging to the AI domain. AI-KG describes 5 types of entities namely methods, metrics, materials, tasks and others linked by 27 relations. AI-KG is a rich source of world knowledge and is designed to support various intelligent services for analyzing and making sense of research dynamics, supporting researchers in their day-to-day work, and informing the decision of founding bodies and research policymakers. In this work, AI-KG is used as a source of domain knowledge to infuse task-specific knowledge in the domain-adaptation scenario. AI-KG is generated by using the automatic pipeline mentioned in [34, 35] before doing the transitive closure and linking the entities to *Wikidata* and CSO. The AI-KG dump can be downloaded in *.ttl* format from https://scholkg.kmi.open.ac.uk/.
- **AI-KG-Small**- It is the reduced version of AI-KG that is generated by post-processing of AI-KG by linking the entities to *Wikidata* and CSO. AI-KG-Small contains 12,094 triples, and the purpose of using AI-KG-Small is to study the "quantity vs. relevance" effect of triples for the given dataset.

3.4 Knowledge-Language Model

This section provides in detail the concepts related to K-LM, followed by the framework and implementation of K-LM. At last, the techniques developed to feed the triples in the K-LM are described.

3.4.1 Concepts Related To K-LM

This subsection describes the concepts and terminologies associated with K-LM used throughout the rest of the chapter.

- **Unique Triples**- For a given KG having $\boldsymbol{N}$ triples, $\boldsymbol{U}$ is defined as the triples having a distinct "subject" entity. Inherently, the triples are arranged in key-value pairs where key $\rightarrow s$ and value $\rightarrow \{p, o\}$. Consider the sentence - *"Italy is home of culture and cuisine"*, and a set of triples (Italy, Country, Europe), (Italy,

famous, Pizza), (Italy, famous_for, Roman Architecture), and (Amsterdam, Capital_of, Netherlands). In the process of knowledge injection, when token ***Italy*** is queried, the *look-up* table returns all the triples with the "subject" ***Italy***, and they are organized as follows: ('Italy', *['country, Europe', 'famous, Pizza', 'famous_for, Roman Architecture']*). In this example {***N=4***} & {***U=2***}. Generating a KG aims at holistically covering the scope of the domain. But all the triples contained within the KG are not necessarily useful for a specific dataset. In this context, ***U*** is proposed as the parameter to measure the relevance of the KG to the dataset. Quantitatively, the less the difference between ***N*** and ***U*** is, the more relevant the KG is.

- **Max Entity**- It is the parameter that controls the number of branches that can be associated with the tokens in the input sentence while injecting knowledge. For $\{Max\_Entity = 1\}$ one triple is associated with the corresponding token of the input sentence and for $\{Max\_Entity = 2\}$, two triples will be associated with the same token, and so on. Note that even if $\{Max\_Entity \geq 2\}$, the number of triples associated with a particular token solely depends upon the triples' availability in the KG. Should the number of available triples for a particular token be less than the value of *Max_Entity*, only the available triple(s) will be associated. This trivial case is demonstrated in Figure 3.1. Consider the input sentence in Figure 3.1 and the tokens within; ***graph***, and ***embedding***. It is assumed that only one triple is available for token ***graph*** and more than two triples are available for token ***embedding*** for triples injection. Therefore, for $\{Max\_Entity = 2\}$ the resulting input sentence has only one triple (***graph***, *uses, human scientific creativity*) associated with the token ***graph*** while the token ***embedding*** has two triples associated, i.e., (***embedding***, *represents, vectors in high-dimensions*) and (***embedding***, *are, compressed representation*). However, a sentence tree can have multiple branches, i.e., several values of *Max_Entity*, but its depth is fixed to one, which means the entity names in triples will not further derive branches iteratively. For our experiments, *Max_Entity* $\in \{1, 2, 3, 4, 5\}$ is used.

3.4.2 Architecture of K-LM

Implementation of K-LM is a two-stage process and is executed by the `K-LM Triple Selection` and `K-LM Classification Modules`. The architecture of K-LM is presented in Figure 3.1. The objective of the `K-LM Triple Selection Module` is to filter and rank (if applicable) the triples provided in AI-KG or AI-KG-Small through a well-defined pipeline. The objective of the second module is to use the triples selected through the `K-LM Triples Selection Module` for injecting

knowledge into the input sentences and then performing the classification task on the provided dataset. The `K-LM Classification Module` of K-LM takes inspiration from the LM model proposed in [76], which was used for binary classification tasks, such as the classification of online reviews for books, hotels, and shopping in the Chinese language. As mentioned before, K-LM is the combination of GPT-2 and BERT. The GPT-2 is used in the `K-LM Triples Selection Module` for ranking the triples, while BERT is used in the `K-LM Classification Module`. The central idea of this work is to use the available world knowledge, independent of a particular experimental dataset; therefore, GPT-2 is not used for generating triples for this work because that will lead to knowledge injection based on the used dataset.

1. **K-LM Triples Selection Module-** This module performs the selection, filtering, and ranking of the triples, which are further used in the `K-LM Classification` module, for knowledge injection. The pipeline of the `K-LM Triple Selection` module as shown in Figure 3.1 is comprised of four sub-modules which are explained below.

    - **Input Triples-** This sub-module is provided with the randomly distributed triples from the input KGs, i.e., AI-KG or AI-KG-Small.
    - **Triples Categorization-** This sub-module performs categorization for generating input triples for the *Triples Injection Methods*. Four distinct categories of triples are produced by this sub-module, namely *Uni-sub-triple*, *Bi-sub-triple*, *Tri-sub-triple*, and *Full KG*[13]. While *Full KG* contains all the triples of the input KG, *Uni-sub-triple*, *Bi-sub-triple*, and *Tri-sub-triple* are categorized based on the $\boldsymbol{L}$ of the input triples.
    - **Triples Injection Methods-** This sub-module implements two methods, *Forward* and *Reverse Injection* for triples selection. The output of this sub-module consists of disordered triples. The *Forward Injection* method employs four techniques: *Uni-sub-triple Injection*, *Bi-sub-triple Injection*, *Tri-sub-triple Injection* and *Full KG Injection*. The *Reverse Injection* method uses the *Reversion Full KG Injection* technique to select the triples. At this stage, the disordered triples have two possibilities:
      a. They are either injected directly into the `K-LM Classification Module` or
      b. They are sent to the sub-module *Triples Ranking* to generate the ordered triples.
      This scenario leads to the development of two approaches, namely Non-deterministic and Deterministic in the pipeline, representing the uniqueness

[13]Refer to Table 3.1 for the definition of *Uni-sub-triple*, *Bi-sub-triple*, *Tri-sub-triple* triples and *Full KG*.

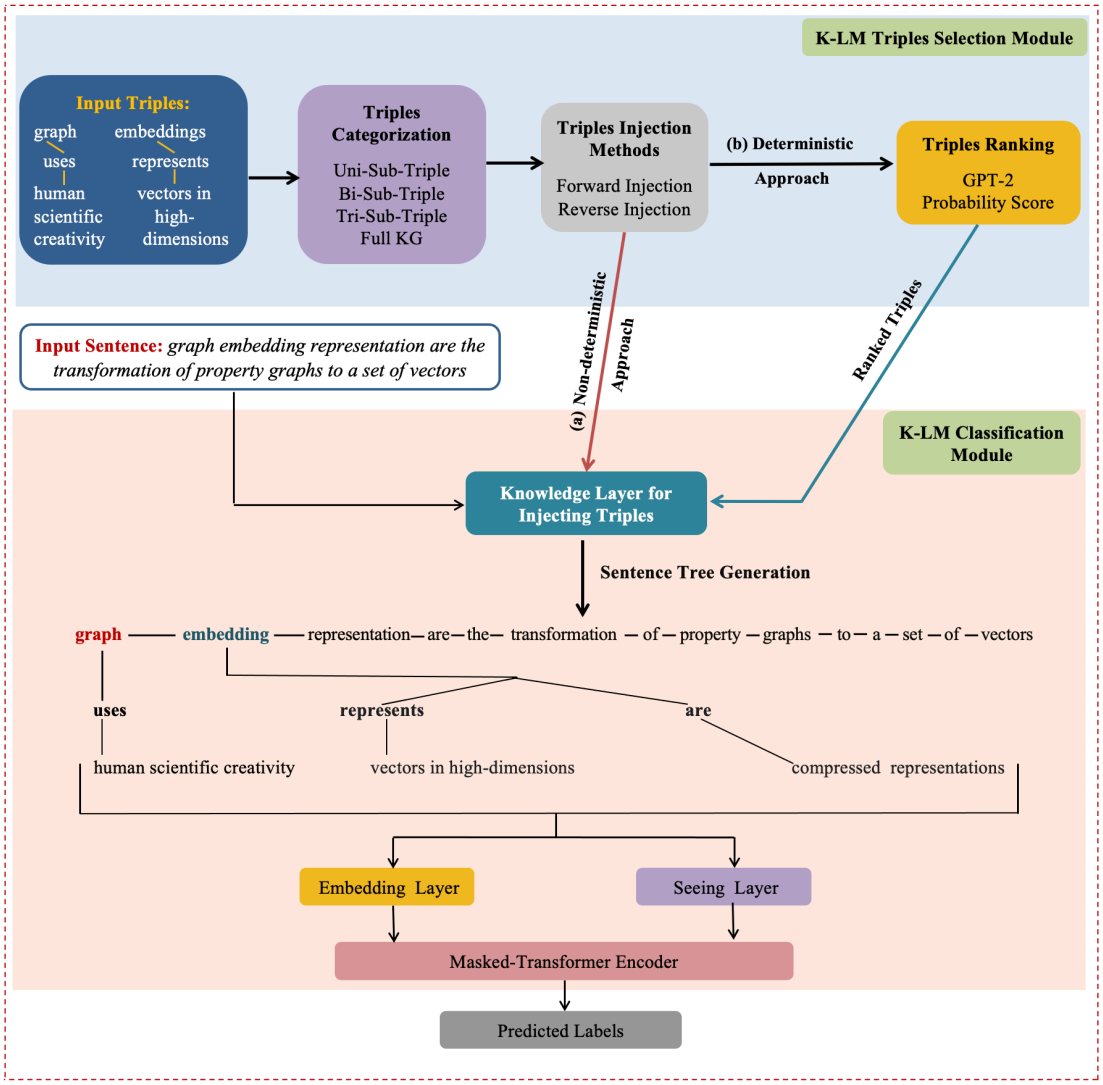


Figure 3.1: K-LM architecture that contains two modules (a) K-LM Triples Selection and (b) K-LM Classification module. Module (a) performs the selection and categorization of the triples and ranks them by employing Non-deterministic and Deterministic approaches. Further, the processed triples are used as the source of domain knowledge in module (b) for the classification task.

and added advantage of K-LM. For ease of understanding, the *Forward* and *Reverse Injection* and the `Non-deterministic` and `Deterministic` approaches are explained separately in subsection 3.4.3.

- **Triples Ranking-** The ranking of triples finds extensive uses when the experimental dataset is small and the input KG is sparsely related to the given dataset. Therefore, this sub-module serves the purpose of infusing the triples based on the context of input sentences for optimal domain knowledge integration. This sub-module takes the disordered triples as input from the

Triples Injection Methods and outputs the ranked triples. The process of ranking the triples is explained below in detail. The triples within the KGs (AI-KG and AI-KG-small) are stored in the *look-up* table and are fetched to the sub-module *Triples ranking*. *GPT2LMHeadModel*[14] is used to rank the triples, which is the GPT-2 Model transformer with a language modeling head on top, and the GPT-2 `Tokenizer`[15]. The GPT-2 model has about 1.5 billion parameters trained on a dataset of 8 million web pages, which makes it very useful to predict the most likely word by interpreting a given sequence of words. In our case, for the given set of triples and context, GPT-2 takes as input each triple iteratively and inserts it in the input sentence providing a score for each sentence thus formed[16]. The outputs are the scores for each sentence used with each triple, known as probability score ***P***. The value of ***P*** is related to the suitability of the underlying triple in the given context of the sentence. More precisely, for a given input sentence $i_s = \{w_o, w_1, w_2, ..., w_n\}$, where $\{w_i\}$ are the words of i_s and the set of triples $t = \{w_0, p_0, o_0\}, \{w_0, p_1, o_1\}, \{w_0, p_2, o_2\}$; the score function is given by $f_s(i_s, t_0^n) \rightarrow \{P_0, P_1, ..., P_n\}$, where $P_0, ..., P_n$ are the probability scores for each triple $\in$ *t*. Hence, in the above-mentioned set of triples *t*, $\{w_0\}$ represents the common subject (and is also present in i_s) while $\{p_0, p_1, p_2\}$ and $\{o_0, o_1, o_2\}$ are the respective predicates and objects of the three triples. In this case, the score function f_s thus returns three probability scores (for each of the three triples having a common subject) $\boldsymbol{P}_0, \boldsymbol{P}_1$, and $\boldsymbol{P}_2$. Based on the obtained probability scores of the sentence, the triples are ranked (ordered) in decreasing value of relevance, i.e., the first triple in the ranking is the best fit for the input sentence. The relevance of triples for the input KG is measured by ***TIT***. The higher the value of ***TIT*** is, the more relevant the KG is to the dataset.

2. **K-LM Classification Module**- It consists of four elements, i.e., knowledge layer for injecting triples, embedding layer, seeing layer, and masked-transformer encoder. For an input sentence, the *Knowledge Layer for Injecting Triples* first injects the triples processed from the `K-LM Triples Selection Module`. The triples are injected in the form of (*s, p, o*). The injection of triples transforms the input sentence into a sentence tree equipped with domain knowledge. The sen-

[14] https://huggingface.co/transformers/model_doc/gpt2.html#gpt2lmheadmodel
[15] https://huggingface.co/transformers/model_doc/gpt2.html#gpt2tokenizer
[16] https://github.com/huggingface/transformers/issues/1009

tence tree thus generated is simultaneously fed to both the embedding and the seeing layers. The embedding layer generates the embedding representation of the flattened input sentence tree, while the seeing layer generates a visible matrix. The visible matrix contains the semantic information of the actual input sentence and provides control over the extent of mutual interaction between the tokens within the input sentence. This mechanism prevents the deviation and false semantic changes from the actual meaning of the original sentence by minimizing the semantic interference and KN caused by triples injection. K-LM is built on top of the BERT (transformer) model; therefore, it uses the same token, position, and segment embedding approach to generate the embedding representation. The difference between the embedding layer of K-LM and BERT is the input type, i.e., K-LM takes sentence tree as input instead of token sequences. Further, the cumulative outputs of embedding and seeing layers are fed to the masked-transformer encoder (which is a modified BERT model in this case), that executes the binary classification task at hand, i.e., predict each input data point as "paper belongs to AI-domain" or not.

3.4.3 Triples Selection Techniques

The core idea of the `K-LM Triple Selection Module` is to provide the `K-LM Classification Module` with selected triples for knowledge integration. This subsection looks at the `K-LM Triple Selection Module` at the fine-grained level and provides further details about the internal mechanism of its sub-modules; *Triples Injection Methods* and *Triples Ranking*. The selection of triples follows the pipeline presented in Figure 3.2, where the input triples are first passed to *Triples Categorization*. In this sub-module, the triples are categorized based on the size $\boldsymbol{L}$ of the triples. The categorized triples are further processed in the sub-module *Triples Injection Methods* following *Forward* and *Reverse Injection*. These two methods uncover the behavior of K-LM in terms of integrating triples. the techniques developed for the optimal selection of triples *Forward* and *Reverse Injection* methods and their interdependencies with the other sub-modules are explained below.

1. **Forward Injection**- Inherently K-LM injects the triples based on the "sequence of occurrence" of words/tokens in the input sentence for a given set of input triples. For instance, consider the input sentence *"**graph embedding representation** is the transformation of property of graphs to a vector or a set of vectors"* and the set of triples:

   (a) *<**graph**, uses, human scientific creativity>*

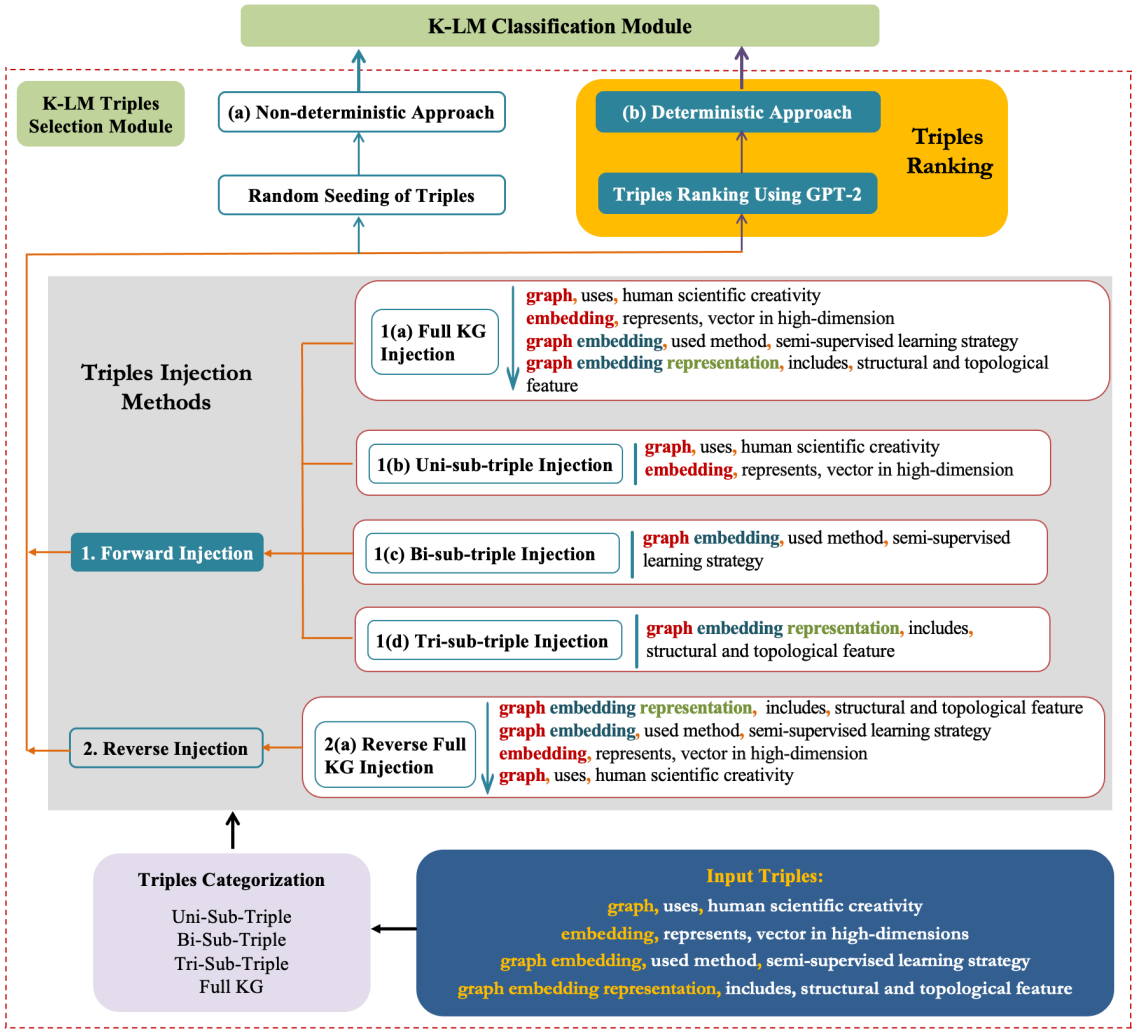


Figure 3.2: Fine-grain demonstration of K-LM Triples Selection Module. The module performs the categorization of triples and further uses them in (a) Forward and (b) Reverse Injection methods.

(b) *<**embedding**, represents, vector in high-dimensions>*

(c) *<**graph embedding**, uses, semi-supervised learning strategy>*

(d) *<**graph embedding representation**, incorporates, structural and topological feature>*

In the knowledge injection process, when the K-LM queries the tokens of the input sentence in the *look-up* table, it first takes into account the token ***graph*** followed by ***embedding*** based on their "sequence of occurrence". Since the triples with the subject ***graph*** and ***embedding*** are already available as input triples, the K-LM automatically integrates these two triples in the sentence. The injection of these triples does not leave any further possibility to inject the triples with sub-

jects < ***graph embedding*** > or < ***graph embedding representation*** >. This is the default behavior of K-LM in which the priority of triples decreases through *Uni-sub-triple*, *Bi-sub-triple* and *Tri-sub-triple* triples. In simple words, the triples having the subject comprised of unigram get the highest priority. The advantage of this method is that it ensures the injection of the highest number of *Uni-sub-triple* triples because the existing number of *Uni-sub-triple* triples in AI-KG and AI-KG-Small is far more than *Bi-sub-triple* and *Tri-sub-triple* triples. The injection of the highest number of *Uni-sub-triple* triples is not always the optimal solution for incorporating domain knowledge, and it is obvious that the *Bi-sub-triple* and *Tri-sub-triple* triples capture more context and meaning. Therefore, having them at a lower priority can make the knowledge injection process prone to KN in the presence of general-purpose KG. Hence, to harness the maximum contextual knowledge from the KGs in the knowledge injection process, four distinct heuristic techniques are designed within the *Forward Injection* method, namely *Full KG Injection Uni-sub-triple Injection*, *Bi-sub-triple Injection*, and *Tri-sub-triple Injection* to mitigate the impact of KN. These four techniques are explained below:

(a) **Full KG Injection**- When the entire KG (containing *Uni-sub-triple, Bi-sub-triple*, and *Tri-sub-triple* triples) is used as the input source for the knowledge injection, it is termed as *Full KG Injection* method. In this method, the triples are injected in the order as follows: *Uni-sub-triple*, *Bi-sub-triple*, and then *Tri-sub-triple* triples, i.e., based on increasing ***"L"*** as shown in Figure 3.2. Therefore, this technique of injecting triples prioritizes the triples so that the injection pipeline first gets *Uni-sub-triple*, then *Bi-sub-triple*, and finally *Tri-sub-triple* triples for injection. Prioritizing the triples here refers to the selection of specific triples from the used KGs.

(b) **Uni-sub-triple Injection**- It is the process of injecting only the *Uni-sub-triple* triples by filtering the *Uni-sub-triple* triples from the KGs used for the experiments.

(c) **Bi-sub-triple Injection**- It is the process of injecting only the *Bi-sub-triple* triples by filtering the *Bi-sub-triple* triples from the KGs used for the experiments.

(d) **Tri-sub-triple Injection**- It is the process of injecting only the *Tri-sub-triple* triples by filtering the *Uni-sub-triple* triples from the KGs used for the experiments.

2. **Reverse Injection**- The opposite process of the *Forward Injection*, i.e., prioritizing the triples in the order *Tri-sub-triple*, *Bi-sub-triple*, and *Uni-sub-triple* is named

Reverse Injection. This method takes *Reverse Full KG* as input that has exactly the opposite priority order as compared to the *Full KG*. The downside arrows in Figure 3.2 corresponding to *Full KG Injection* and *Reverse Full KG Injection* show the decreasing priority of the triples. The advantage of this method is that it firstly selects *Tri-sub-triple* triples, which are semantically rich in this case, for injection and thus substantially reduces the number of triples required for enhancing K-LM's performance.

Considering the examples (input sentence and the set of triples) used in the *Forward Injection* method; Figure 3.3 illustrates the sentence tree structure thus generated after employing the *Forward* and *Reverse Injection* methods using the *Full KG* as input triples. The vital point to note here is that *Forward* and *Reverse Injection* methods are the mandatory steps through which the triples have to pass and they are independent of the next sub-module *Triples ranking*. Depending upon the type of input triples required for knowledge injection, i.e., disordered or ordered (ranked) and are explained below.

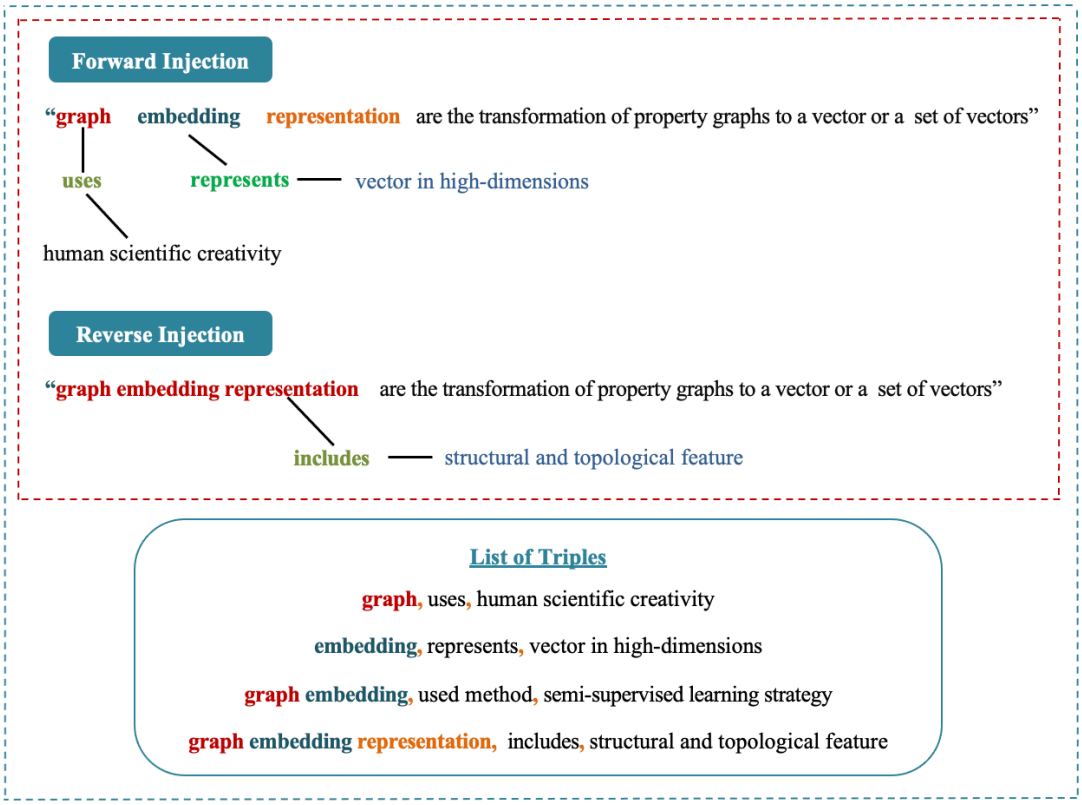


Figure 3.3: Sentence tree formation after using Forward and Reverse Injection methods for the listed triples. The priority order of triples in Forward Injection is *Uni-sub-triple* > *Bi-sub-triple* > *Tri-sub-triple* and vice-versa for the Reverse Injection.

1. **Non-deterministic Approach**- The triples received from the sub-module *Triples Injection Methods* are selected through the techniques as shown in Figure 3.2. At this stage, their distribution in the *look-up* table remains random and they are independent of the "context" of the input sentences. Therefore, due to the varying order of triples' arrangement, this random seeding of triples is termed as `Non-deterministic` approach. In this approach, the triples are injected directly into the `K-LM Classification Module`.

2. **Deterministic Approach**- On the other hand, another possibility is to send the triples to the sub-module *Triples Ranking*. If this option is chosen, *Triples Ranking* ranks the input triples based on the probability score using GPT-2. Further, the ranked triples are organized in the *look-up* table using the ranking, and this ordered arrangement of triples is constant throughout the fine-tuning and inference stage. *Triples ranking* finds extensive uses when the experimental dataset is small and KG is sparsely related to the given dataset. This approach is termed as `Deterministic` and it is "context-dependent".

In other words, the above two methods can also be looked at as context-aware and context-unaware approaches to seed triples in the knowledge injection process. The rationale behind developing these methods is finding a trade-off between "relevance of KG" and "computational complexity" and maximizing the generalization of triple's injection. Hence, while the `Deterministic` approach gives an edge over the number of triples injections required to enhance the classification model's performance, randomly seeding the triples is more aligned to make the knowledge injection process independent of the dataset.

3.5 Experiments and Results

The computational resource used to develop our methods and perform the experiments is mentioned in Table 2.2. The `K-LM Classification Module` of K-LM takes inspiration from the LM model proposed in [76] and the source code is available at Git-hub[17].

3.5.1 Experiments

In this paper, experiments are conducted within the *Scholarly Domain* to tackle a binary classification task by using the KGs AI-KG and its variant AI-KG-Small. The experiments performed can be summed up as a combination of approaches as below mentioned:

[17] https://github.com/vsrana-ai/K-LM

- **Ranking the triples**- `Deterministic` and `Non-deterministic` approaches are used to perform the experiments with *Max_Entity* $\in \{1, 2, 3, 4, 5\}$.
- **Forward and Reverse Injection**- The experiments with five methods of knowledge injection namely, *Full KG Injection*, *Uni-sub-triple*, *Bi-sub-triple*, *Tri-sub-triple*, and *Reverse Full KG Injection* are performed. Here *Full KG Injection* is the default mode of seeding the triples, while the remaining four techniques proposed in this work allow to custom feeding the triples.
- **Knowledge resources**- As the source of domain knowledge, AI-KG and AI-KG-Small are used to inject the additional knowledge to perform all the experiments.
- **Bert Base**- When K-LM is not provided with any KG, then it is equivalent to a general-purpose pre-trained BERT model. Hence, in this paper, the experiment `"K-LM Without KG"` signifies the baseline approach of the BERT model.
- **Baselines**- For the baseline comparison, Naive Bayes and Random Forest classifiers are employed as CML and a two BiLSTM layers-based network with pre-trained GLoVe[18] and fastText[19] word embedding as DL approaches.

3.5.2 Experimental results

This sub-section presents the results obtained from the experiments conducted in this work. To make all the employed classification approaches comparable; the same distribution of test, validation, and train sets of Scholarly dataset is used, which are 60%, 20%, and 20% respectively. The baseline approaches have tackled the problem of binary classification using CML and DL approaches which do not use any additional domain knowledge, i.e., is independent of KGs. The results of the baselines of CML and DL approaches along with K-LM are summarized in Table 3.2.

Table 3.2: Performance of CML and DL approaches with Scholarly dataset

Approach	Accuracy	F-1 Score
Naive Bayes (22,000 Features)	59.60	58.93
Random Forest (26,661 Features)	74.21	62.45
Bi-LSTM (Pre-Trained GloVe)	73.78	73.23
Bi-LSTM (Pre-Trained fastText)	74.88	73.48
BERT Base	78.12	75.05
K-LM using GPT-2 (Deterministic)	**81.98**	**79.55**
K-LM random seeding (Non-deterministic)	**81.78**	**79.50**

[18] https://nlp.stanford.edu/projects/glove/
[19] https://fasttext.cc/docs/en/english-vectors.html/

Tables 3.3 and 3.4, summarize the results of five `Non-deterministic` modes of triples injection with *Max_Entity* acquiring values from 1 to 5. Tables 3.5 and 3.6, summarize the results of five `Deterministic` modes of triples injection with *Max_Entity* acquiring values from 1 to 5. Finally, Tables 3.7 and 3.8 summarizes the comparative distribution of ***N***, ***U***, and the actual number of triples injected for each approach used in the experiments.

Table 3.3: Performance of K-LM on the Scholarly dataset with AI-KG-Small (Non-deterministic triples seeding) using the proposed five modes of triples injection

Experiment	Full KG Injection		Uni-sub-triple Injection		Bi-sub-triple Injection		Tri-sub-triple Injection		Reverse Full KG Injection	
Max Entity	**Acc.**	**F-1**	**Acc.**	**F-1**	**Acc.**	**F-1**	**Acc.**	**F-1**	**Acc.**	**F-1**
1	74.49	71.55	**79.53**	**76.60**	**81.78**	**79.50**	**81.82**	**79.45**	**81.07**	**79.11**
2	72.74	66.90	79.23	75.60	81.13	78.85	81.77	79.05	81.05	78.80
3	71.37	66.75	77.75	73.60	81.12	78.65	81.52	78.80	80.72	77.75
4	71.22	65.60	77.44	74.00	81.02	77.60	81.45	79.45	80.85	78.40
5	70.64	63.80	76.42	72.55	80.52	78.25	81.53	79.35	80.62	72.55

Table 3.4: Performance of K-LM on the Scholarly dataset with AI-KG (Non-deterministic triples seeding) using the proposed five modes of triples injection

Experiment	Full KG Injection		Uni-sub-triple Injection		Bi-sub-triple Injection		Tri-sub-triple Injection		Reverse Full KG Injection	
Max Entity	**Acc.**	**F-1**	**Acc.**	**F-1**	**Acc.**	**F-1**	**Acc.**	**F-1**	**Acc.**	**F-1**
1	70.05	60.80	**79.68**	**76.55**	80.25	**76.90**	80.87	77.75	80.48	77.20
2	69.46	61.40	78.35	74.95	**80.28**	76.80	**81.32**	**78.50**	**80.60**	**77.30**
3	68.68	57.40	77.44	73.35	78.80	74.80	80.80	77.35	79.48	75.50
4	67.34	56.20.	76.09	72.80	78.40	74.20	80.60	77.20	79.48	75.50
5	66.28	55.45	76.10	70.95	77.90	73.60	80.40	77.05	80.12	77.20

3.5.3 Results analysis

In this subsection, insights from the experimental results and in-depth analyses of the outcomes are provided.

- The methods proposed to custom feed the triples into K-LM have shown superior performance and have outperformed the baselines, BERT base, and *Full KG* Injection significantly. The best performance from baselines is observed by BERT Base with an accuracy of **78.12**, while K-LM has outperformed BERT Base with

Table 3.5: Performance of K-LM on the Scholarly dataset with AI-KG-Small (Deterministic triples seeding) using the proposed five modes of triples injection

Experiment	Full KG Injection		Uni-sub-triple Injection		Bi-sub-triple Injection		Tri-sub-triple Injection		Reverse Full KG Injection	
	Acc.	**F-1**	**Acc.**	**F-1**	**Acc.**	**F-1**	**Acc.**	**F-1**	**Acc.**	**F-1**
Max_Entity = 1	75.55	69.85	**80.25**	**76.50**	**81.25**	**78.15**	**81.98**	**79.05**	80.72	**77.90**
Max_Entity = 2	72.79	67.50	79.33	75.00	80.47	77.35	81.82	78.90	**80.80**	77.40
Max_Entity = 3	72.87	67.15	78.70	75.35	80.68	77.30	80.87	77.80	80.05	77.20
Max_Entity = 5	71.92	66.80	77.19	73.50	80.60	77.20	81.68	78.55	80.58	77.40
Max_Entity = 5	71.12	64.80	76.57	72.65	80.45	76.80	81.52	78.45	79.93	76.95

Table 3.6: Performance of K-LM on the Scholarly dataset with AI-KG (Deterministic triples seeding) using the proposed five modes of triples injection

Experiment	Full KG Injection		Uni-sub-triple Injection		Bi-sub-triple Injection		Tri-sub-triple Injection		Reverse Full KG Injection	
	Acc.	**F-1**	**Acc.**	**F-1**	**Acc.**	**F-1**	**Acc.**	**F-1**	**Acc.**	**F-1**
Max_Entity = 1	70.47	66.15	**80.67**	**78.20**	**80.95**	**78.50**	80.98	78.15	**81.35**	**79.20**
Max_Entity = 2	70.12	64.12	78.65	75.30	80.22	77.20	81.58	79.55	80.43	77.75
Max_Entity = 3	69.89	59.10	77.44	73.85	79.62	76.85	81.30	79.55	80.7	78.30
Max_Entity = 5	68.22	57.31	76.72	73.30	79.13	76.00	81.42	79.15	80.52	77.65
Max_Entity = 5	67.47	54.22	76.37	71.55	78.93	75.55	**81.72**	**79.25**	80.07	77.25

Table 3.7: Triples distribution of AI-KG-Small for each experiment with Scholarly dataset.

KG	**Full Kg**	**Uni-sub-triple (Top-1000)**	**Bi-sub-triple**	**Tri-sub-triple**	**Reverse Full KG**
AI-KG-Small (Total)	12904	2027	1719	325	2148
AI-KG-Small (Unique)	2397	113	401	82	498
AI-KG-Small (Injected)	360	113	324	66	408

Table 3.8: Triples distribution of AI-KG for each experiment with Scholarly dataset.

KG	Full Kg	Uni-sub-triple (Top-1000)	Bi-sub-triple	Tri-sub-triple	Reverse Full KG
AI-KG (Total)	1877453	37387	402845	85206	273096
AI-KG (Unique)	735884	201	35876	11248	40862
AI-KG (Injected)	16695	201	23753	7263	25575

all the five knowledge injection methods; the highest accuracy being **81.98** when used with *Tri-sub-triple Injection*.

- Except for *Tri-sub-triple Injection* the remaining four techniques observe a consistent decline in the model's accuracy when *Max_Entity* increases from 1 to 5. There are two reasons behind this. First, *Tri-sub-triple* triples are very less in number as compared to *Uni-sub-triple* and *Bi-sub-triple* triples, which highlight the unique association of *Tri-sub-triple* triples with the context of the input sentence. Secondly, *Tri-sub-triple* triples are semantically rich and they capture more context as compared to *Uni-sub-triple* and *Bi-sub-triple* triples. So, when more *Tri-sub-triple* triples are injected, they relate to a large part of the input sentence and enhance the semantics of the input sentence with relevant knowledge that helps the classifier to predict classes. On the other hand, injecting too many *Uni-sub-triple* and *Bi-sub-triple* triples increases the chances of introducing irrelevant knowledge along with domain knowledge in the given context and hence makes the classifier prone to KN. The KN is inversely proportional to the model's accuracy; hence, K-LM's accuracy decreases with the increase in KN. Therefore, it can be concluded that $\{Max\_Entity = 1\}$ is the optimal value for knowledge injection when *Uni-sub-triple* and *Bi-sub-triple* triples are used.

- The experimental findings show that in the **"quantity vs. relevance"** of triples, the relevance of triple is of utmost importance, and it directly influences the K-LM's performance. It can be understood by the fact that $\{Max\_Entity = 1\}$ allows the injection of only one triple per token of the input sentence. For such an arrangement of input sentences in the sentence tree, the size of KGs becomes irrelevant as the performance of K-LM becomes dependent on ***U***, ***TIT*** and its relevance to the semantic context. The results mentioned in the Tables 3.7 and 3.8 validate the propositions:
  a) too much knowledge injection makes the K-LM prone to KN.
  b) KGs with a low $\boldsymbol{N}/\boldsymbol{U}$ ratio are a better fit for knowledge injection, and so is the

AI-KG-Small for the experiments in this work.

c) *Tri-sub-triple* Triples injection requires significantly less number of triples injection, and that directly relates to less training time and resource requirement to conduct the experiments.

3.6 Conclusion and Future Work

Knowledge injection is a very delicate process; while too much knowledge injection makes the LM prone to noise and leads to false semantic changes, insufficient knowledge hardly improves the LM's efficacy. In this work, the LM, K-LM, and a well-defined pipeline to integrate world knowledge directly in the form of triples from available world knowledge is presented. In the pipeline, first, methods such as *Forward* and *Reverse Injection* are implemented to select and filter the triples for the knowledge integration process. In the later stage, `Non-deterministic` and `Deterministic approaches` are employed to tackle the "relevance vs. quantity" juxtaposition. The results show the efficacy of the proposed knowledge injection methods, as each of them has significantly outperformed the CML and DL baselines. The results demonstrate that K-LM is a potential choice to solve knowledge-driven tasks by using a few triples and helps in the presence of small training data. This work can be extended to other domains in real-life scenarios to achieve optimal classification models' performances by only using a few triples as a source of external domain knowledge. Since this is the first work towards quantifying the knowledge injection and implementing a pipeline to select and filter the triples, there is further scope for improvement in developing a more intelligent and optimized LM. Some limitations of the current work can be narrowed down to the manual selection of the *Max_Entity* parameter (used to control the number of triples associated per token of input sentences to mitigate the KN), the test of the K-LM for *Scholarly Domain* only, and the usage of one KG for the target domain (more KGs within the scholarly domain should be tested for a wider and more comprehensive analysis). Each of the proposed methods of injecting triples has outperformed the baseline approaches, with no method being a clear winner for all the values of *Max_Entity* $\in \{1, 2, 3, 4, 5\}$, used for the experiments. Therefore, it is an interesting future direction to automate the selection of the *Max_Entity* parameter by taking into account its dependency on the variables, such as Unique Triples of the input KG, context, and text length of input data, for an optimal knowledge injection. To generalize the use of K-LM, the future direction of the work is also set to field-test K-LM beyond the *Scholarly Domain* in NLP downstream tasks [7]. In this regard, the target is mental health and its sub-domains, which are considered complex research domains in healthcare, where the integration of external domain knowledge can lead to increased reliability of the classification models.

I also intend to use other state-of-the-art LR models such as Elmo [104], XLNet [152], etc. Further, I aim to develop novel techniques to generate KGs relevant to the available datasets of given domains to tackle more efficiently the problem of imbalanced datasets.

Part II: Generating Motivational Interviewing Dataset and its Benchmarking Evaluation

Chapter 4

Anno-MI: Generating dataset of counselling therapy

4.1 Introduction

Patient health can be significantly improved by changes in behaviour, such as reducing alcohol consumption [115]. Counsellors, however, may have difficulty convincing patients to adopt such changes. Thus, MI [88] has been developed as an effective counselling approach that evokes motivation for change from the client[1] themselves. Correspondingly, coding systems such as Motivational Interviewing Skill Code (MISC) [87] and Motivational Interviewing Treatment Integrity (MITI) [89] are commonly used to identify MI codes and aspects related to therapist and client. Recent years have seen significant interest in the research of linguistic and statistical MI analysis. The first computational model for identifying reflection, a key skill in MI, was introduced [18]. More broadly, the modelling of MI-related aspects such as codes and therapist empathy has been approached with methods based on classical machine learning [147, 4, 49] (e.g. support-vector machines) and deep learning [48, 148, 47, 19] (e.g. recurrent neural networks). In terms of data resources [102], recently published a corpus of high- and low-quality MI dialogues taken directly from online video-sharing platforms and analysed the linguistic features that capture the differences between high- and low-quality MI.

Despite the progress, NLP for MI has been hindered by the lack of publicly accessible MI dialogue data and annotations, owing to privacy-related restrictions. As research in this field has been conducted primarily on private/undisclosed annotated MI dialogues, it has been challenging to replicate and build on previous findings. Previously, to the

[1] A client receiving therapy may not have an illness, thus the term "client" is used in lieu of "patient" in this work.

best of our knowledge, the only publicly and freely available MI dataset was created by [102], consisting of transcripts of MI videos on YouTube/Vimeo obtained through automatic speech recognition (ASR). However, the transcript quality is compromised by the substantial ASR noise and frequent wrongly assigned interlocutor labels (client utterances labelled as therapist utterances, and vice versa) that cause difficulty in understanding. In the paper [102] the authors have also analysed two MI codes — reflection and question — based on the dataset annotations from trained students, but those annotations are unavailable at the time of writing.

To address the scarcity of publicly available resources for MI-related NLP research and broaden access to this area, we introduce Anno-MI presented in [142], a dataset of 133 MI-adherent[2] and non-adherent therapy conversations that a) are professionally transcribed from MI demonstration videos on YouTube & Vimeo b) are built with explicit consent from the video owners that allows dataset creation, public release and use for research purposes, and c) are annotated on key MI aspects by experienced MI practitioners. In addition to above mentioned, the key contribution of this work is summed up below:

1. Detailed, visualised statistical analyses of the Anno-AugMI to examine its patterns and properties and

2. Two Anno-MI-based utterance-level prediction tasks with potential for real-world applications, and experiment with different machine-learning models as baselines to facilitate comparison with future methods.

3. Performance analysis of Anno-MI-based utterance-level classifiers on different topics as well as their generalisability to new topics.

4.2 Background & Related Work

4.2.1 MI Coding

The gold standard for examining counsellor adherence to therapy protocols is behavioural observation and coding [6], which provides feedback and evaluation of therapy sessions. During the coding process, trained annotators assign labels to therapist skills/behaviours such as reflection and client behaviours such as change talk. Session-level ratings on qualities such as empathy are often also included. A variety of coding schemes have been proposed, including the MISC [87] and the MITI [89]. However, as manual

[2]In this work, "MI-adherent" is used as a synonym of "high-quality" and similarly "MI non-adherent" as a synonym of "low-quality". These terms are not related to video quality or transcription quality.

coding is costly and time-consuming, automatic coding of utterance-level behaviour and related tasks such as the automatic rating of therapist empathy have garnered significant research interest in recent years.

4.2.2 Available Resources

MI conversation resources are scarce. As real-world therapy often contains sensitive topics and information, counselling dialogues are mostly privately owned or proprietary (e.g. therapy transcripts from Alexander Street[3]). As for resources, annotated MI corpora such as [99] have been built from sources such as wellness coaching phone calls and leveraged for tasks like utterance-level code prediction [101] and empathy prediction [100], but they mostly remain publicly inaccessible. To the best of our knowledge, the only freely and publicly available MI corpus to date is [102], created based on automatic transcripts of MI videos on YouTube/Vimeo. The paper [102] also collected annotations with respect to reflections and questions for the corpus and conducted related analyses, but those annotations are not available at the time of writing. Also, considerable ASR noise and wrong interlocutor labels exist in the corpus 4.3.2, thus limiting the quality of the dataset.

4.2.3 Text-Based Approaches to MI Analysis

In terms of text-based approaches to automatic coding, [18] used n-grams and similarity features to develop the first model for identifying reflection, while the work in [4] used a labelled topic model to generate MI codes. More recently, deep-learning-based models have been utilised. For example, studies in [148] and [126] used RNNs for behaviour prediction, followed by [47] who did so under a multi-label multi-task setting to improve the performance as well [19] who also investigated forecasting the codes of upcoming utterances. For therapist empathy modelling, an early approach is from [147] with an n-gram language model. In [49], authors leveraged language features inspired by psycho linguistic norms, while [48] used LSTMs to produce turn-level behavioural acts further processed to predict empathy. Separately [144, 143] explored leveraging links between therapeutic and general-conversation empathy to tackle therapist empathy prediction in low-resource scenarios.

[3] https://alexanderstreet.com/products/counseling-and-psychotherapy-transcripts-series

4.2.4 Speech-Based & Multimodal Methods for MI Analysis

For utterance-level code prediction [122], proposed an LSTM-based [54] end-to-end model that directly predicts codes given speech features without using ASR. Most other works leveraging speech features for code prediction exploit multi-modal features, such as [23] and [123] where LSTMs with joint prosodic and lexical features are utilised. For session-level therapist empathy modelling, more speech-only methods have been proposed, including [146] which studied prosodic features such as jitter and shimmer from speech signals as well as [149] which investigated speech rate entrainment.In addition [45] proposed an automatic rating tool of MI sessions using speech and language features, predicting a range of session-level codes including empathy and MI spirit in addition to utterance-level codes.

4.3 Creating Anno-MI

Considering the scarcity/absence of publicly available conversation datasets of real-life MI and their privacy-related legal and ethical restrictions, the dependency is on demonstrations of MI-adherent and non-adherent therapy from online video-sharing platforms, in a similar vein to [102]. In this work, after obtaining explicit consent from the video owners, professional transcripts of the demonstrations are generated and MI experts are recruited to annotate the transcripts following a scheme covering key MI elements.

4.3.1 MI Demonstration Videos

As a trade-off between therapy authenticity and privacy-related constraints, only MI conversations from online video-sharing platforms (YouTube/Vimeo) are considered. With an exhaustive keyword search (such as “effective MI” and “using MI” for MI-adherence and “ineffective MI” and “bad MI counselling”) and building on [102], 346 demonstrations of MI-adherent and non-adherent therapy were identified. According to the literature on client-centered counselling [88], in a high-quality session, the therapist centers on the client and expresses empathy, while in a low-quality session they mostly provide instruction and suggestions. Each video is labeled as high- or low-quality MI based on its title (e.g. “Motivational Interviewing - Good example”, “The Ineffective Physician: Non-Motivational Approach”) as well as descriptions and narrator comments (e.g. “This is $\cdots$ video $\cdots$ where I demonstrate how to use motivational interviewing $\cdots$”). Those labels are considered to be automatically validated since the video uploaders are professional therapists and established institutions dedicated to positive

Table 4.1: Dataset overview

	High-Quality MI	Low-Quality MI
#Conversations	110 (82.7%)	23 (17.3%)
#Utterances	8839 (91.1%)	860 (8.9%)

Table 4.2: Dialogue excerpts from high- & low-quality MI where the goal is smoking cessation/reduction. **Therapist**: therapist; **Client**: client.

High-Quality MI
Therapist: Mm-hmm. So it's kind of surprising to you that something you've been doing and you've been doing more and more of it is actually pretty bad for you. **Client**: Oh, yes. I checked the box on your form when you asked if I use tobacco, I checked "No" because I never thought of myself as a smoker. **Therapist**: Mm-hmm. What do you kind of make of that now that you realize that you're actually a tobacco user and that you might actually be causing some pretty serious health effects?
Low-Quality MI
Therapist: So you're gonna quit then? **Client**: Uh, maybe. **Therapist**: What do you mean, maybe? I just told you how bad it is for you. It's messing up your mouth, you're putting yourself at risk for all these other diseases. This is really important. You need to quit.

behaviour change. To generate Anno-AugMI, the video owners are contacted for obtaining their explicit consent[4] to use their videos to create, analyse and publicly release the transcript-based MI dialogue dataset. Explicit permissions are obtained to use 119[5] of those videos, which contained 133 complete conversations — a video may contain multiple dialogues. 110 of the dialogues showcase high-quality MI and the other 23 low-quality MI (Table 4.1). A pair of high- and low-quality MI session excerpts, both about smoking cessation/reduction, are presented in Table 4.2. The imbalance with respect to high- and low-quality MI dialogue volumes can be attributed to a) fewer low-quality MI video owners responding to the request or consented; b) low-quality MI videos are relatively scarce on Youtube/Vimeo, possibly because MI-adherence demonstrations are deemed more valuable as "good examples" and thus filmed and uploaded more.

[4]The consent of the individuals in the videos was gathered together with that of the content owner where applicable.

[5]42 of the 119 videos are overlapped with [102].

4.3.2 Transcription

Using a professional transcription service[6], fluent and faithfully transcribed MI conversations were collected from the videos, whereas the transcripts of [102] were produced by automatic captioning. While a step of verifying video content-caption matching is reported in [102], in practice, it was found that a considerable number of incorrectly transcribed words/phrases and mismatched interlocutor (therapist/client) labels exist in the corpus of [102] that can significantly hinder text understanding. Table 4.3 presents the excerpts from [102] and Anno-MI of the same video to exemplify the marked difference in transcription quality between the two datasets. Anno-MI is also free from other noises such as narrations but retains context-relevant details, including "hmm", "right" and interlocutor sentiment/emotion [111, 41, 110] indicators such as "[laugh]".

4.3.3 Expert Annotators & Workload Assignment

As MI annotation requires specialised knowledge, only experienced MI practitioners are engaged to annotate the transcripts. Specifically, 10 therapists found through the Motivational Interviewing Network of Trainers[7], an international organisation of MI trainers and a widely recognized authority in MI, are recruited for the task. All the annotators had high proficiency in English and prior experience in practicing and coding MI. 7 transcripts with different lengths and MI qualities are assigned to every expert to facilitate inter-annotator agreement (IAA) computation, while each of the other 126 transcripts is randomly assigned to exactly one expert. Overall, each expert annotated 19 to 20 transcripts with total lengths of around 144 minutes in terms of the total duration of the original 19 to 20 videos. The 7 IAA transcripts are about 45 minutes in length in total, and no expert was aware that those transcripts would be used to compute the IAA. The IAA results of Anno-MI are not directly comparable with those of other annotated MI corpora, since the former is calculated based on the annotations from 10 experts while the latter often come from much fewer (e.g. 2 or 3) annotators, and it is usually less likely to reach the same or higher level of IAA with more annotators. This also means that the attributes of Anno-MI that do have good IAAs are indeed reliably annotated.

4.3.4 Anno-MI and "Real-World" MI

For Anno-MI to be useful for real-world applications, it is crucial that its dialogues reflect both high- and low-quality MI in the real world. Therefore, the annotators af-

[6] https://gotranscript.com/
[7] https://motivationalinterviewing.org/trainer-listing

Table 4.3: Transcription quality comparison between Anno-MI and Pérez-Rosa et.al. Color code to locate the differences are: incorrectly transcribed word (red); omitted words/phrases(blue); words from the other interlocutor that should have started a new utterance (orange); missing client/therapist utterance (cyan).

Anno-MI
Client: Right. Well, it would be good if I knew, you know, that my kids are taken care of too- **Therapist**: Yeah. **Client**: so I'm not worried about them while I'm at work. **Therapist**: Right. Yeah. Because you're- you're the kind of parent that wants to make sure your kids are doing well. **Client**: Right. **Therapist**: Yeah. Um, so tell me, what would it take to get you to like a five in confidence, to feel a little bit more confident about getting work? **Client**: Well, I mean, being able to make the interviews would be the priority. **Therapist**: Okay, Yeah. **Client**: Um, so chi- you know, taking care, having some childcare, having- **Therapist**: Mm-hmm. **Client**: - having someone I trust that I can call when I know I've got an interview. **Therapist**: Yeah. Because you definitely need to go to an interview in order to get the job. **Client**: Right. Yeah. **Therapist**: So having taken care of that part, having some reliable childcare would definitely help. **Client**: Yeah.
[102]
Client: one it would be good if I knew you know that my kids are taking care of ("too") - yeah so I'm not worried about them law in the work right yeah **Therapist**: because you're you're the kind of parent that wants to make sure your kids are doing well great ({C}) yeah um so tell me what would it take to get you to like a five in confidence to feel a little bit more confident ("about") getting work **Client**: well I mean being able to make the interviews would be the priority again ({T}) um so try you know taking care having some child care I mean having ({T}) someone I trust that I can call when I you know what that interview because you definitely need to go to an interview ~~of~~ in order to get ~~three~~ ("the job") **Therapist**: ~~yeah~~ yeah so having taken care of that part having some reliable child care ("would definitely help") **Client**: yeah definitely not

ter they completed their tasks, were asked whether they felt the Anno-MI dialogues resembled real-world MI or else. As shown in Figure 4.1, 83% of the responses "agree" or "somewhat agree" that the therapist utterances and the dialogues overall reflect real-

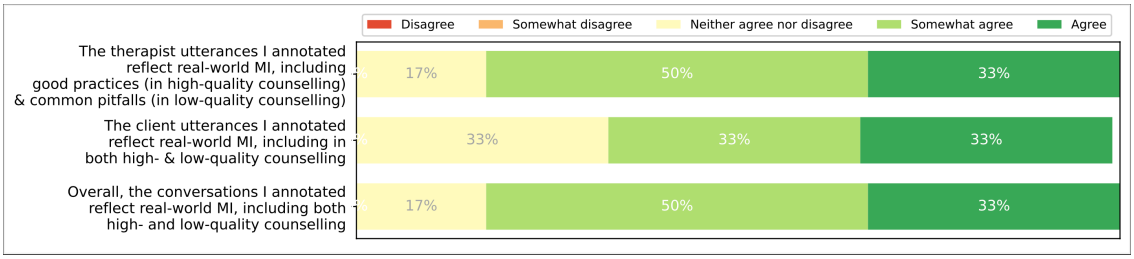


Figure 4.1: Results of survey for annotators regarding whether Anno-MI reflects real-world high- and low-quality MI.

world MI, and the figure is 66% for the client utterances. The clear majority in each case shows that Anno-MI indeed sufficiently captures the characteristics of real-world MI, even though the dialogue sources are demonstrations. It is noted that researchers, especially those in corporate environments, are faced with a very challenging legal and regulatory landscape in the field of NLP for counselling, due to privacy-related concerns and rules in different jurisdictions. Therefore, a dataset like Anno-MI can be used significantly more broadly, since it does not have any privacy implications or legal issues concerning different jurisdictions.

4.4 Annotation Scheme

To generate Anno-MI, a detailed annotation scheme is designed to study therapist and client behaviours, based on the MI literature, existing coding protocols (MISC/MITI), and feedback from therapists. At the conversation level, the annotators were asked to briefly describe the dialogue goal, e.g., "smoking cessation". Thus, in Table 4.4 the top 10 topics are summarized in terms of a) the number of conversations that have those topics, and b) the total number of utterances in those conversations. At the utterance level, the annotation scheme is shown in Table 4.5. Each annotator was asked to annotate all utterance-level attributes and to select only one from the range of labels for each attribute. When annotating an utterance, an annotator could also see the preceding and subsequent utterances for more contextual reference.

4.4.1 Therapist Utterance Attributes

(Main) Behaviour

Asking, Informing and Listening are three basic but important communication skills in MI that enable efficient and effective counselling [115]. Based on this principle and relevant parts of mainstream MI coding systems, **Question**, **Input**, and **Reflection** are established as three major therapist behaviours for analysing Asking, Informing and Lis-

Table 4.4: Top 10-topics in Anno-MI in terms of 1) number (percentage) of conversations that have those topics, and b) total number (percentage) of utterances in those conversations.

Topic	**#Dialogues**
Reducing alcohol consumption	28 (21.1%)
Smoking cessation	21 (15.8%)
Weight loss	9 (6.8%)
Taking medicine / Following medical procedure	9 (6.8%)
More exercise / Increasing activity	9 (6.8%)
Reducing drug use	8 (6.0%)
Reducing recidivism	7 (5.3%)
Compliance with rules	5 (3.8%)
Asthma management	5 (3.8%)
Diabetes management	5 (3.8%)
Other	33 (24.8%)
Topic	**#Utterances**
Reducing alcohol consumption	1914 (19.7%)
Reducing recidivism	1303 (13.4%)
Smoking cessation	1106 (11.4%)
Diabetes management	709 (7.3%)
Reducing drug use	578 (6.0%)
Taking medicine / following medical procedure	574 (5.9%)
More exercise / increasing activity	525 (5.4%)
Weight loss	396 (4.1%)
Avoiding DUI	394 (4.1%)
Changing approach to disease	315 (3.2%)
Other	2107 (21.7%)

tening, respectively. In cases where more than one behaviour is present in an utterance, e.g. a question after input, the expert is asked to select the **main behaviour**. **Other** as a fourth behaviour is also considered in this design, where no **Question**, **Input**, or **Reflection** is shown in the utterance. **Question**, **Input** and **Reflection** are listed as separate attributes of therapist utterances in order to investigate their sub-types, as laid out in the sections below. It is to note that this work is more focused on the use of Asking, Informing and Listening in the Anno-MI dialogues[8], therefore it does not seek to compare directly with previous work that uses the complete MISC/MITI for annotation.

[8]For the same reason, the original annotation scheme was more ambitious and had several non-MITI/MISC annotation fields, but they are not included in this paper due to their very low IAAs (Fleiss' kappa) and the space limit, and this is why the annotation scheme presented in this section may look like a subset/regrouping of MISC to some readers.

Table 4.5: Utterance-level multi-choice annotation scheme. **(+)** implies presence of utterance attribute (e.g. "Simple reflection" entails that **Reflection** exists in utterance), while **(-)** indicates absence thereof (e.g. "No reflection" label implies **Reflection** is not present in utterance).

Therapist Utt. Attrib.	**Label**
(Main) Behaviour	**Question** **Input** **Reflection** **Other**
Question	Open question **(+)** Closed question **(+)** No question **(-)**
Input	Information **(+)** Advice **(+)** Options **(+)** Negotiation/Goal-Setting **(+)** No input **(-)**
Reflection	Simple reflection **(+)** Complex reflection **(+)** No reflection **(-)**
Client Utt. Attrib.	**Label**
	Negative
Talk Type	Change Neutral Sustain

Question

Therapists use Asking to develop an understanding of the client and their problems. Therefore, we include **Question** as a therapist behaviour and define any question as *open* or *closed* in accordance with mainstream MI coding conventions. The definition of open/closed is similar to that of open-ended/closed-ended questions except for some nuanced differences (e.g. "Tell me more about it." is considered an open question). Some examples are given in Table 4.6[9].

Input

Informing is the primary manner of communicating knowledge to the client. To include a wide range of conveyed knowledge, the term **Input** is used and list *advice*, *information*,

[9]All the labelling examples are for illustration purposes and are not actual dialogues from the dataset.

Table 4.6: Example Labelling for therapist **Question**

Utterance	Question Type
Did you use heroin this week?	Closed
On a scale from 1 to 10, how motivated are you to quit?	Closed
How do you feel about that?	Open
Tell me about your smoking.	Open

Table 4.7: Example Labelling for therapist **Input**

Utterance	Input Type
Your blood pressure was elevated when the nurse took it this morning	Information
You could try this respiration exercise to calm down when you're anxious	Advice
Do you want to stay where you're at, quit, or cut down?	Options
Would it be doable for you to cut down on your smoking by 2 packs of cigarettes?	Negotiation/Goal-setting

giving options and *negotiation/goal-setting* as its types. Some examples are given in Table 4.7. When an utterance contains more than one type of **Input**, the annotators choose the "main" type of **Input** to make the labels mutually exclusive and facilitate utterance-level NLP applications.

Reflection

A crucial way of Listening is reflective listening, as it shows listening, hearing and understanding the client and can thus be effective in helping people change (lead by change talk). Following MISC, two reflection types are considered in this annotation scheme: *simple & complex*. Simple reflection conveys an understanding of what the client has said but with little additional meaning, e.g. repeating. In comparison, complex reflection shows a deeper understanding of the client's point of view and adds substantial meaning to their words, using techniques such as metaphors, exaggeration, and summary [115]. A pair of simple & complex reflections of the same client utterance are shown in Table 4.8.

Table 4.8: Example Labelling for therapist **Reflection**

Interlocutor	Utterance	Reflection Type
Client	At one time I was pretty much anti anything but marijuana	
Therapist 1	Marijuana was OK	Simple
Therapist 2	That's where you drew the line	Complex

Table 4.9: Example Labelling for client **Talk Type**

Utterance	Talk Type
My doc told me I'm gonna lose my leg if I don't start checking my blood sugars	Change
I hate a night without a buzz	Sustain
Uh huh	Neutral

4.4.2 Client Utterance Attributes

Talk Type

As pointed out in MI literature [115], clients usually feel ambivalent about adopting positive behaviour change, and thus an essential objective of MI is for clients to convince themselves to change if it is compatible with their personal values and aspirations. Such talks for change are known as "change talks". Conversely, "sustain talks" show resistance to change and a desire to preserve the status quo. Finally, "neutral talks" indicate no preference for or against change. Hence, *change talk*, *sustain talk* and *neutral talk* are used in this work as the three types of the client **Talk Type** attribute. Table 4.9 presents some examples of those talk types.

4.5 Inter-Annotator Agreement (IAA)

4.5.1 Default Measure: Fleiss' Kappa at Utterance-Level

Fleiss' kappa [43] is used as the default measure for calculating utterance-level inter-annotator agreement (IAA) over the annotations on the 7 transcripts. Considering 3 ways of calculation: **All**, **All**(Strict), and Binary. **All** applies to all the utterance attributes, while the other two modes apply to **Input**, **Reflection** and **Question** only.

Specifically, since those three attributes have a default "absence" option (i.e., No input, No reflection and No question, as shown in Table 4.5); a two-class presence-vs.-absence (i.e. BINARY) IAA is computed for them in addition to the fine-grained all-class IAA (i.e. **All**). When computing **All**-IAA for **Question**, for example, we consider the original label space: {Open question **(+)**, Closed question **(+)**, No question **(-)**}, where **(+)** means there is a question in the utterance and **(-)** means there is not. Conversely, only the presence-vs.-absence {**(+)**, **(-)**} space is considered when calculating BINARY-IAA.

All(STRICT)-IAA, is also calculated which computes IAA within the original label space but on a more challenging subset of utterances, motivated by the observation that it is substantially easier to distinguish between the presence **(+)** labels than between presence **(+)** and absence **(-)**. For example, differentiating between "Simple reflection **(+)**" and "Complex reflection **(+)**" is harder than between **Reflection** and non-**Reflection**. Therefore, we compute **All**(STRICT) on the utterances where at least one annotator chose a presence **(+)** option. For **Reflection**, for example, we calculate **All**(STRICT)-IAA on the utterances where at least one annotator selected "Simple reflection **(+)**" or "Complex reflection **(+)**".

4.5.2 Results of Default IAA Measure

All Fleiss'-kappa-based IAAs are listed in Table 4.10. Following [69], the IAAs are grouped as slight (0.01-0.20), fair (0.21-0.40), moderate (0.41-0.60), substantial (0.61-0.80) and almost perfect (0.80-1.00) agreement scale. An attribute is considered **predictable** if its IAA shows moderate or better agreement. It is observed that the utterance attributes where BINARY and **All**(STRICT) are applicable, the order of agreements is, without exception, **All**(STRICT)-IAA $<$ **All**-IAA $<$ BINARY-IAA, which proves the challenge of the subset used for computing **All**(STRICT)-IAA as well as the ease of annotating the absence/presence of a particular utterance attribute. The annotators are found to be in fair agreement on **Input** (**All**(STRICT)) and **Reflection** (**All**(STRICT)), which reveals the difficulty of annotating those attributes despite their inclusion in MISC/MITI, particularly when their presence in an utterance cannot be easily ruled out. Nevertheless, the IAA jumps to a substantial agreement for **Input** and **Reflection** under the BINARY setting, which suggests the presence of distinguishable linguistic features unique to those two attributes. Encouragingly, **Question**, **(Main) Behaviour** and **Talk Type** all record moderate or better IAAs under all settings, which shows the text-based predictability and therefore the existence of distinct linguistic features of those attributes.

Based on the IAA results above, the attributes are kept with their respective label spaces that show moderate or better IAAs in the release version of the enhanced AnnoMI:

Table 4.10: Inter-annotator agreements on utterance-level annotations, in Fleiss' kappa. Color code: Orange, Blue, Cyan and Green indicate fair (0.21-0.40), moderate (0.41-0.60), substantial (0.61-0.80) and almost perfect (0.80-1.00) agreement, respectively.

Therapist Utterance Attribute	**IAA Setting**	**IAA**
Input	**All**(STRICT)	0.34
	All	0.51
	BINARY	0.64
Reflection	**All**(STRICT)	0.32
	All	0.50
	BINARY	0.66
Question	**All**(STRICT)	0.54
	All	0.74
	BINARY	0.87
(Main) Behaviour	**All**	0.74
Client Utterance Attribute	**IAA Setting**	**IAA**
Talk type	**All**	0.47

- **Input**: BINARY - {with input, without input}). **N.B.** Namely, each therapist utterance has this label indicating whether **Input** is present. The same applies to **Reflection**.
- **Reflection**: BINARY - {with reflection, without reflection})
- **Question**: **All**, i.e., {open question, closed question, no question}),
- **(Main) Behaviour**: **All**, i.e., {Reflection, Input, Question, Other})
- **Talk Type**: **All**, i.e., {Change, Neutral, Sustain}

For the 7 IAA transcripts, the value of each attribute of each utterance is obtained through majority voting.

4.5.3 Supplementary IAA Measure: Intraclass Correlation

Following MITI, Intraclass Correlation (ICC) is also used to analyse **(Main) Behaviour** and **Talk Type** at label-level to gain more insights and facilitate comparison with other studies. For each label, the number of occurrences of utterances annotated with the label in each session by each annotator is counted. Thus, each of the 10 annotators has 7 label counts corresponding to the 7 IAA transcripts. Then, ICC is computed to describe how much of the total variation in the label counts is due to differences among

Table 4.11: Inter-annotator agreements as Intraclass Correlation.

(Main) Therapist Behaviour	ICC
Input	0.975
Reflection	0.991
Question	0.997
Other	0.996
Client Talk Type	ICC
Change	0.916
Neutral	0.986
Sustain	0.890

annotators. Also following MITI, the ICC scores are obtained using a two-way mixed model with absolute agreement and average measures.

As Table 4.11 presents, all the **(Main) Behaviour** and **Talk Type** labels have excellent (0.75-1) [26] agreement scores, which shows the reliability of Anno-MI annotations. Nevertheless, Change and Sustain have slightly lower ICCs — around 0.9 — compared to the other ICCs that are almost 1.0, which somewhat echoes the lower Fleiss'-kappa-based IAA of **Talk Type** compared to that of **(Main) Behaviour**.

4.6 Dataset Analysis

The annotations are analysed via visualisations, unless otherwise specified, **(Main) Behaviour** represents the behaviour of an utterance. For example, if a therapist's utterance consists of a reflection and a question but **Reflection** is annotated as the main behaviour, the utterance to be a reflection is considered instead of a question, in order to facilitate further analysis. It is noted that while there are clear correlations between utterance attribute distribution and MI quality in some cases, they do not necessarily point to causation, especially given the relatively low amount of data and potential sampling bias.

4.6.1 General (Main) Behaviour and Talk Type Distributions

As Figure 4.2 demonstrates, the most marked contrast between therapist behaviours in MI-adherent and non-adherent therapy is the proportions of **Reflection** and **Input**. The average MI-adherent therapist employs **Reflection** in 28% of their utterances whereas it is only 7% in non-adherent therapy, echoing the MI requirement of trying to understand the client's perspective and communicating it. On the other hand, **Input** is given 33% of the time in low-quality MI but only 11% in high-quality MI, which, together

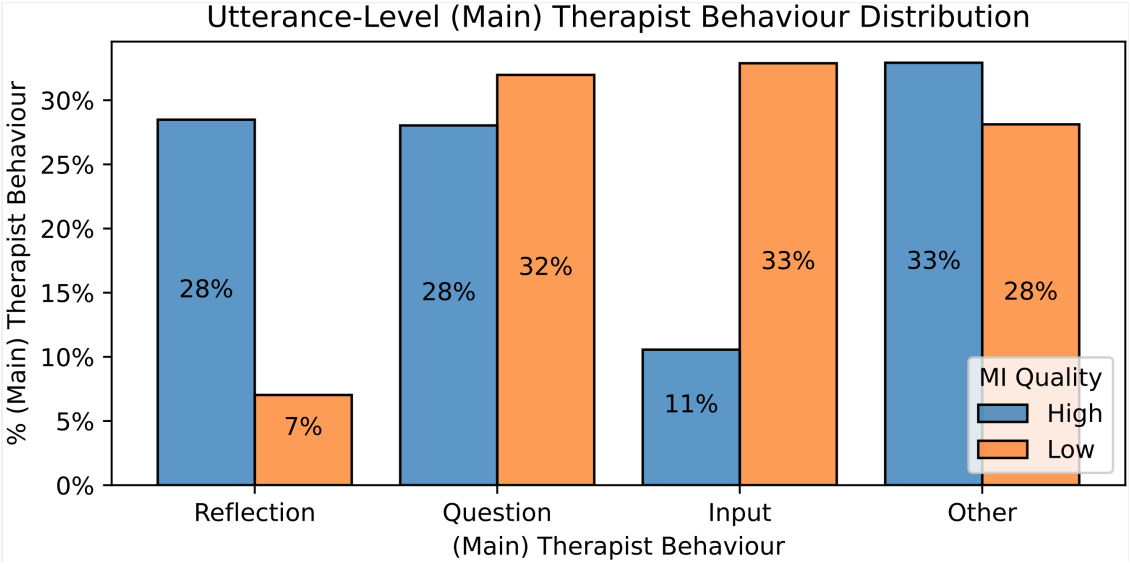


Figure 4.2: (Main) Behaviour distributions in high- & low-quality MI

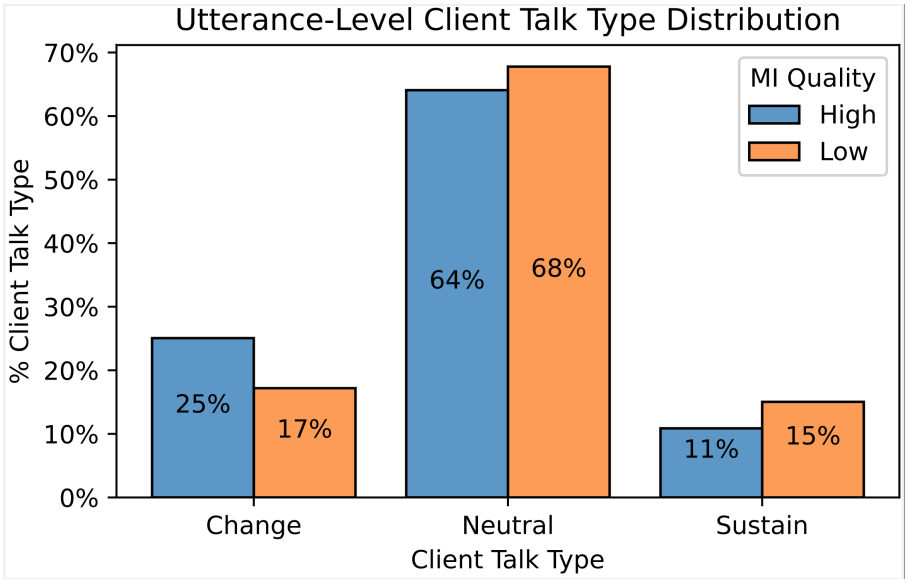


Figure 4.3: Talk Type distributions in high- & low-quality MI

with the statistics of **Reflection**, conforms to the observation [115] that high-quality MI emphasises understanding the client as opposed to speaking from their own point of view. The correlation between MI quality and the share of **Question** and **Other** is relatively weak.

As for **Talk Type**, change talk is more frequent in high-quality MI — 25% vs. 17%, whereas sustain talk has a stronger presence in low-quality MI — 11% vs. 15% (Figure 4.3). Those contrasts are, nevertheless, less obvious than those found in **Reflection** and **Input**. Possible explanations include a) some clients in low-quality MI could adopt tepid change-talk-like speech such as "Yeah, maybe" only to end the counselling quickly

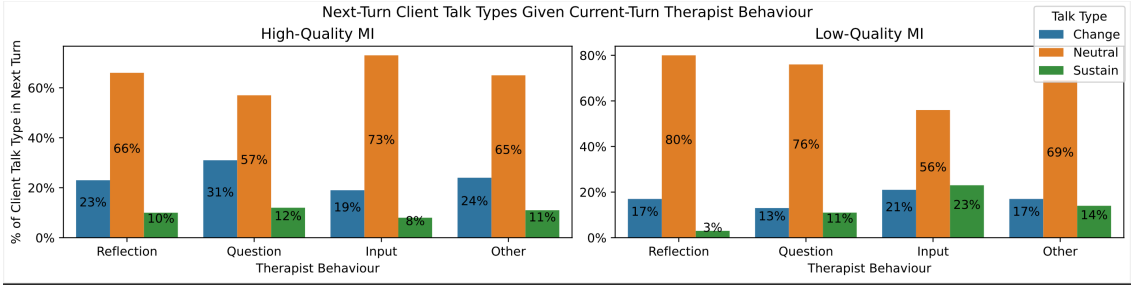


Figure 4.4: Distribution of next-turn client talk types given different therapist behaviours in the current turn

and b) some clients in high-quality MI are simply more reluctant to change but the therapist still respects that, as is recommended in MI. On the other hand, most (64%-68%) client utterances belong to the neutral talk category regardless of MI quality, for which the prevalence of short utterances like "Mhmm" and "Uh huh" can be a major contributing factor.

4.6.2 Posterior (Main) Behaviour and Talk Type Distributions

MI guidelines have specific recommendations on how a therapist should respond when the client talks in certain ways, and a client may also react to the therapist in particular patterns. The posterior distributions of next-turn therapist behaviours(/client talk types) is probed given the current-turn client talk type(/therapist behaviour). Denoting u_t^T as the therapist utterance at turn (time step) t and u_{t+1}^C as the client reply in the following turn, the posterior distribution of client talk types can be represented as $p(\textit{Talk\_Type}(u_{t+1}^C) \mid \textit{Behaviour}(u_t^T))$. Similarly, the posterior distribution of therapist behaviours can be formulated as $p(\textit{Behaviour}(u_{t+1}^T) \mid \textit{Talk\_Type}(u_t^C))$. Figure 4.4 presents the posterior distribution of client talk types (i.e. $p(\textit{Talk\_Type}(u_{t+1}^C) \| \textit{Behaviour}(u_t^T))$).

While neutral talk is clearly the majority talk type of the client response, in most cases $p(\textit{Talk\_Type}(u_{t+1}^C) = \text{Change} \| \textit{Behaviour}(u_t^T))$ is substantially larger in high-quality MI than in low-quality MI regardless of $\textit{Behaviour}(u_t^T)$, which shows that an MI-adherent counsellor is more likely to evoke change talk from the client, irrespective of specific therapist behaviours. On a more granular level, **Question** is the most likely (31%) therapist behaviour in high-quality MI to evoke change talk, which may be because some therapist questions lead to change talks more often, such as asking the client what steps they could take towards a behaviour change or how confident they are about adopting a change. Interestingly, **Input** results in more change talks (21%) than any other therapist behaviour in low-quality MI, but it is also the therapist behaviour that prompts the most (23%) sustain talks, which may suggest that the effect of frequent input — characteristic of low-quality MI as shown in Figure 4.2 — is far from certain in

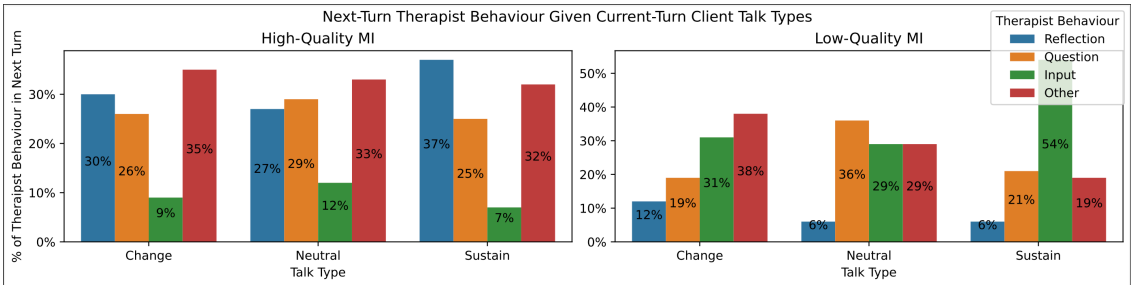


Figure 4.5: Distribution of next-turn therapist behaviours given different client talk types in the current turn

terms of evoking change talk and reducing sustain talk. Figure 4.5 shows the posterior distribution of therapist behaviours (i.e. $p(\textit{Behaviour}(u_{t+1}^{T}) \| \ \textit{Talk\_Type}(u_t^{C}))$).

One can observe that MI-adherent therapists in general use considerably more reflections than non-adherent therapists do — 30% vs. 12% — in response to change talks, which suggests that high-quality MI utilises **Reflection** to reinforce willingness to change. On the other hand, the most commonly shown therapist behaviour in response to sustain talk in high-quality MI is **Reflection** (37%), while the dominant pattern of reacting to sustain talk in low-quality MI is **Input** (54%). This contrast serves as strong evidence that MI-adherent therapy focuses more on showing empathy and trying to understand the client when faced with resistance, including through **Reflection**, whereas a non-adherent therapist is more likely to try to challenge, correct or persuade the client through more **Input** — a common mistake in MI non-adherent therapy [115].

4.6.3 (Main) Behaviour and Talk Type as Conversation Proceeds

Following [102], each conversation is divided into 5 parts: $[0.0, 0.2]$, $(0.2, 0.4]$, $(0.4, 0.6]$, $(0.6, 0.8]$ and $(0.8, 1.0]$, in order to probe conversational properties at different dialogue stages. Specifically, the distributions of different therapist behaviours and client talk types are examined at those stages. Among the trends shown in Figure 4.6[10], one can observe in both high- and low-quality MI that the proportion of **Question** gradually decreases as the therapist gathers more information about the client from the progressing conversation. The amount of **Reflection**, on the other hand, generally fluctuates within a small interval throughout a dialogue in both high- (27% - 31%) and low-quality MI (2% - 8%), which means **Reflection** is common throughout a high-quality MI session and rare throughout a low-quality one. Finally, the proportion of **Input** rises during the middle stages ($(0.4, 0.8]$) in both high- and low-quality MI, but the increase is substantially more pronounced in low-quality MI sessions (from ∼30% to ∼60%) than in

[10]In all the line charts, the "marked" data points are the sample means and the error bars around them are calculated using bootstrapping with a 95% confidence interval.

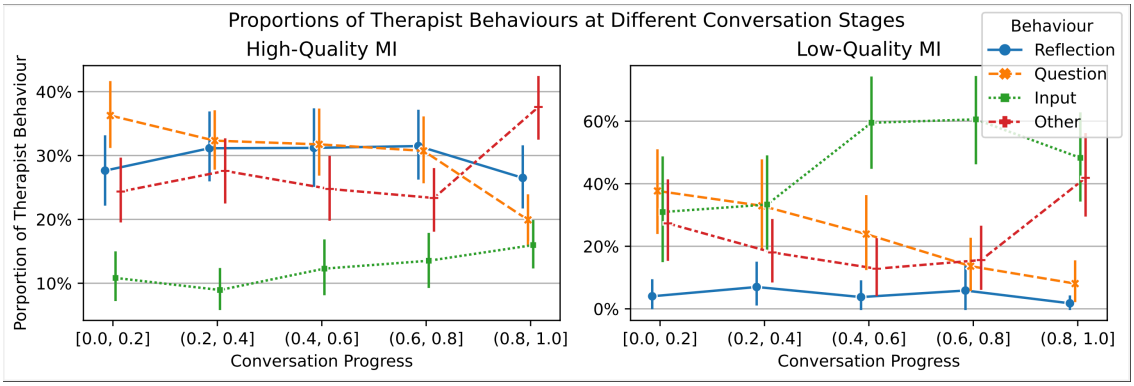


Figure 4.6: Proportions of therapist behaviours in different conversation stages in high- and low-quality MI.

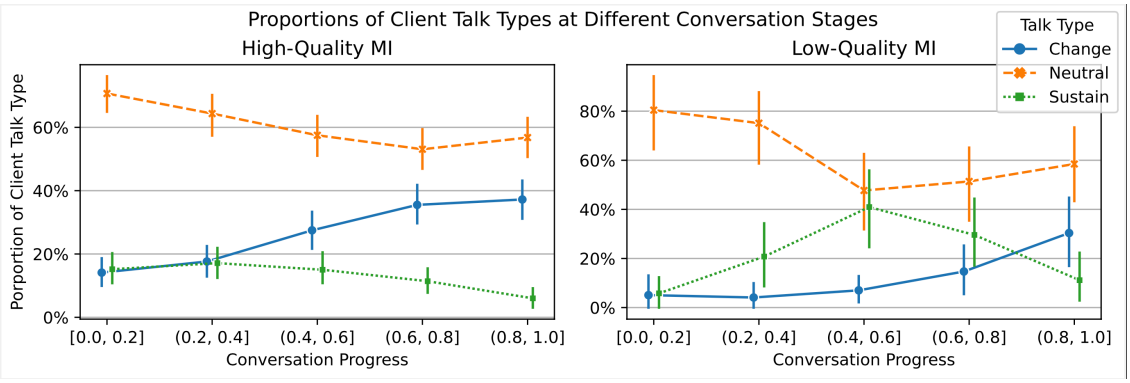


Figure 4.7: Proportions of client talk types in different conversation stages in high- and low-quality MI.

high-quality ones (from ∼10% to ∼15%), which further indicates a non-adherent therapist tends to talk from their own perspective more as the conversation develops.

The trends of different client talk types are displayed in Figure 4.7. A clear shift is shown in high-quality MI: there are similar amounts of change and sustain talk at the beginning of a conversation, but change talk becomes more present steadily and eventually reaches around 40% at the end of a dialogue, while the share of sustain talk diminishes gradually at the same time and drops to around 7%. In other words, the desired effects of MI-adherent therapy, namely change talk evocation and sustain talk reduction, become increasingly prominent with the progress of a session. In low-quality MI, however, during the early & middle conversation stages (i.e. $[0.0, 0.6]$) the proportion of sustain talk soars from approximately 10% to a little over 40% while the number for change talk remains under 10%. Interestingly, the later stages (i.e. $(0.6, 1.0]$) show the opposite trend, as the growing share of change talk surpasses the declining proportion of sustain talk, finishing at around 30% and 12% respectively at the end. Nevertheless, the absolute %**Change** - %**Sustain** difference is clearly larger at the end of a high-quality

MI session in general.

4.7 Utterance-Level Prediction Experiments

From the annotation labels, various utterance-level prediction tasks can be readily defined. This section focuses on two tasks: **therapist behaviour prediction** and **client talk type prediction**. These tasks are introduced as examples of potential real-world applications of Anno-MI, in order to inspire future tasks based on the dataset. While an imbalance exists between the high- and low-quality dialogue volumes, its impact on the tasks is expected to be minor, since they are not related to MI quality directly. For future work exploring session- or utterance-level MI quality classification, however, remedies such as data augmentation will be needed to address the imbalance. Each task allows a single utterance as the input and requires a class label as the output. Experiments are performed with 4 machine-learning-based models trained until convergence. In addition the BERT variants with AdapterHub[11] [105] (in turn based on HuggingFace [140]), the CNN models with Keras [12], and the other models with Scikit-learn [97] are implemented.

- **BERT w/o Adapters**: BERT-base-uncased [39] fine-tuned on Anno-MI.
- **BERT w/ Adapters**: BERT-base-uncased with adapters [55, 105] fine-tuned on Anno-MI. Adapters are a small set of task-specific parameters that can be easily plugged into transformer [132]-based models so that only the lightweight adapters are updated during fine-tuning while the rest of the model is frozen.
- **CNN**: convolutional neural networks initialised with word2vec embedding [83] and fine-tuned on Anno-MI.
- **Random Forest**: random forest with TF-IDF features.

Two dummy baseline classifiers are also used for comparison:

- **Prior**: producing random predictions based on the class distribution in the training set.
- **Uniform**: producing random predictions based on the uniform distribution of the classes.

[11] https://github.com/Adapter-Hub/adapter-transformers
[12] https://keras.io/

Considering the relatively small size of Anno-MI, 5-fold cross-validation (CV) on the entire dataset is applied. **Matthews Correlation Coefficient (MCC)** [80] is used as the metric to leverage its robustness to class imbalance. MCC ranges between -1 to 1, where -1 represents total disagreement between prediction and observation, 0 means no better than a random prediction, and 1 indicates perfect prediction. For Prior and Uniform whose outputs are random, the models are run 1000 times in each training-validation setup and average their performances. Therefore, each of the 6 models listed above eventually has 5 performances from a 5-fold CV, and the mean is taken as the final performance of the model.

To address the class imbalance, two variants for each training set are introduced: **Original Unbalanced** and **Augmented Balanced**. The former keeps the original data in each CV training set, while the latter leverages a Pegasus [155]-based neural paraphraser[13] in order to augment the non-majority classes so that the size of each class in **Augmented Balanced** reaches that of the majority class in **Original Unbalanced**. Table 4.12 presents both the multi-class and per-class performance of each model, the former measured by multi-class MCC and the latter by binary MCC. To calculate the latter for a particular class, each ground truth and predicted label is converted to True or False depending on whether the label is the same as the class in question.

4.7.1 Task 1: Therapist Behaviour Prediction

We first investigate **therapist behaviour prediction**. Given a therapist's utterance, the task is to predict its (main) therapist's behaviour. Overall, the BERT variants score the highest with MCCs around 0.75, followed by CNN at 0.6 and Random Forest at approximately 0.5. Compared to the random baselines (Prior & Uniform) with MCCs at 0 which confirms the randomness of their prediction, the trained models, especially the BERT variants, have clearly learned contextualized semantics. No substantial difference exists between the results of BERT w/o Adapters and BERT w/ Adapters. The effects of augmentation are minor and mixed, as the technique slightly improves the performance of Random Forest and CNN and harms that of the BERT variants marginally. The order of difficulty for classifying individual therapist behaviours is generally **Other** $<$ **Question** $<$ **Reflection** $<$ **Input**. Remarkably, the performance difference between **Other** and **Input** is 0.24 $\sim$ 0.28 MCC even for the BERT variants, suggesting **Input** is relatively challenging to classify. Also, the difference between the performance on **Other** — mostly short utterances like "Hmm" and "OK" — and that on the other three behaviours is substantially larger on the non-BERT models than on the BERT variants, likely attributable to the strong context modelling capability of BERT.

[13] https://huggingface.co/tuner007/pegasus_paraphrase

Table 4.12: Overview of the multi-class and per-class performance of (main) therapist behaviour prediction and client change talk type prediction. All results are averaged from 5-fold cross-validation. ↓/↑ indicates decrease/increase from the original-data-trained model's performance to that of the augmented-data-trained model.

Result Format	Result of Model Trained on Augmented Data and Original Data				
	(Main) Therapist Behaviour Prediction				
Model	*Multi-Class*	**Other**	**Question**	**Reflection**	**Input**
BERT_{adpt}	.74 (.74)	.84 ↑ (.82)	.78 ↓ (.80)	.68 (.68)	.56 ↓ (.58)
BERT	.74 ↓ (.75)	.84 (.84)	.79 (.79)	.68 ↓ (.69)	.56 ↓ (.58)
CNN	.60 (.60)	.80 (.80)	.59 ↑ (.58)	.52 ↑ (.50)	.41 ↑ (.39)
Random Forest	.50 ↑ (.49)	.76 ↑ (.75)	.40 (.40)	.38 ↑ (.37)	.38 ↑ (.34)
Prior	.00 (.00)	.00 (.00)	.00 (.00)	.00 (.00)	.00 (.00)
Uniform	.00 (.00)	.00 (.00)	.00 (.00)	.00 (.00)	.00 (.00)
	Client Talk Type Prediction				
	Multi-Class	**Change**	**Neutral**	**Sustain**	
BERT_{adpt}	.34 ↓ (.36)	.34 ↓ (.36)	.36 ↓ (.39)	.29 ↑ (.27)	
BERT	.32 ↓ (.37)	.32 ↓ (.37)	.34 ↓ (.40)	.29 ↑ (.28)	
CNN	.24 ↓ (.26)	.23 (.23)	.26 ↓ (.31)	.21 (.21)	
Random Forest	.22 ↑ (.19)	.23 ↑ (.21)	.24 ↑ (.21)	.15 ↑ (.08)	
Prior	.00 (.00)	.00 (.00)	.00 (.00)	.00 (.00)	
Uniform	.00 (.00)	.00 (.00)	.00 (.00)	.00 (.00)	

4.7.2 Task 2: Client Talk Type Prediction

Client talk type prediction aims to produce the right client talk type label given a client utterance. Overall, this task records universally lower scores than Task 1 for all the trained models — the best BERT-variant performances are around 0.35 MCC while CNN and Random Forest score about or less than 0.25, irrespective of data augmentation. Two factors likely responsible for the performance gap between the two tasks are **dialogue context** and **annotation noise**. In some cases, the talk type of a client utterance can only be determined with context grounding. For example, "Yeah" as a reply to "So you work out every day?" is a neutral talk, but it should be change talk when it follows "Don't you ever wish things were different?". Also, the IAA (Fleiss' kappa) for client talk type is around 0.47 while it is 0.74 for therapist behaviour, which suggests that annotating talk type is more challenging and therefore more noise is present in the labelling. Inevitably, such noise makes it harder to optimise the trainable models. Among the talk types, Neutral Talk has slightly higher performance than Change Talk, while Sustain Talk is more challenging ($\Delta \geq 0.05$ MCC) than both for all the classifiers, which is somewhat unexpected, considering the similar IAA scores for Change Talk and Sustain Talk. As mentioned previously, using more dialogue context may offer a performance boost and more insights, for now it is left for future work. In this ex-

perimental setup the single-utterance BERT variants is established as strong baselines for both tasks, in order to facilitate comparison with improved machine-learning-based methods in the future.

4.8 Topic-Specific and Cross-Topic Performance

Apart from performance over the entire Anno-MI, we also explore how the models fare in conversations of different topics, hypothesising that some topics may be more challenging for certain models on particular tasks. Importantly, as the generalisability to topics unseen during training is a major desideratum of reliable models with real-world impact, we probe cross-topic model performance by training on data of all but one topic and testing on examples from that topic. Based on the topic coverage of Anno-MI (Table 4.4), three topics are selected namely — reducing alcohol consumption, reducing recidivism, and smoking cessation — for probing the topic-specific and cross-topic performance of all the trained models on the two tasks defined in 4.7, since between 10% and 20% of the utterances in Anno-MI belong to conversations of these topics. All the results (in MCC) are summarised in Table 4.13.

4.8.1 Topic-Specific Performance

To obtain the performance on topic T_i, we re-use the 5-fold CV models for the two tasks (4.7), but we test each model only on a T_i-specific subset of the corresponding 20%-Anno-MI test set created during CV. Specifically, the subset consists entirely of utterances that are originally from conversations of topic T_i. By averaging the performances of the 5 models on their respective T_i-specific subsets, this method covers all T_i-utterances and thus yields a reliable measure of the T_i-specific performance of each model type. Generally, it is clear that the model performances, especially those of the BERT variants, follow the topic-wise ordering below, with negligible impact from class balance/imbalance:

- **Therapist Behaviour Prediction:** reducing alcohol consumption $>$ smoking cessation $>$ reducing recidivism
- **Client Talk Type Prediction:** reducing alcohol consumption $\approx$ smoking cessation $>$ reducing recidivism

One contributing factor to the topic-level performance gaps could be the coverage of the three topics — reducing alcohol consumption $>$ reducing recidivism $>$ smoking cessation, as better coverage entails more data used for training. However, it is also

clear that the client talk type prediction performance for reducing-recidivism conversations is considerably lower — , e.g., 0.15 by BERT w/ ADAPTERS — than for smoking cessation — , e.g., 0.37 by BERT w/ ADAPTERS, despite the slightly larger coverage of reducing recidivism. Such a contrast is, therefore, more likely because the utterances of the topic themselves are more semantically challenging for the task, and it also shows the necessity to include a wide range of topics in a counselling dialogue dataset.

4.8.2 Cross-Topic Performance

It is often important for trained models to generalize to unseen domains. While conversations of different topics are not completely different domains, the results shown in 4.8.1 illustrate that models indeed have varying levels of performance depending on the topic. Hence, to complement 4.8.1 where models trained on dialogues of **all** topics are examined for their topic-specific performances, we probe model generalizability by removing a topic T_i from the training set completely and then analyzing its performance on a T_i-only test set. Concretely, we adopt a leave-1-topic-out approach by training on all the Anno-MI utterances from conversations that do **not** have topic T_i and testing on all the Anno-MI utterances from dialogues that **only** have topic T_i. Conversations with T_i, as well as a different topic, are not present during training or testing. We note that the test set in this setup is effectively identical to that of 4.8.1; therefore, the cross-topic and topic-specific performances can be compared fairly. Unsurprisingly, cross-topic (trained on leave-1-topic-out data) performance is lower than its topic-specific performance counterpart (i.e., topic-specific 5-fold CV) for most <model type, topic> combinations (Table 4.13), since the models have limited exposure to the left-out topic during leave-1-topic-out training whereas all topics are present during the training of the CV models. There is not a clear trend as to which topic generally leads to the largest gap between cross-topic and topic-specific performance, and the impact of augmentation-enabled class balance on prediction performance is mixed. Encouragingly, the BERT models only see minor performance degradation (≤ 0.02) and even slight improvements in some cases for therapist behaviour prediction, thus illustrating the generalisability of those models for this task. For client talk type prediction, however, larger performance drops (≥ 0.04) are more common. For example, BERT w/ ADAPTERS sees a fall to 0.29 in cross-topic from 0.38 in topic-specific when the topic is reducing alcohol consumption under the setting of balanced training data. Considering that the overall performances in 4.7 of this task are lower than those of therapist behaviour prediction, one may postulate that client talk type prediction is more challenging (when conversation history is absent in the input), and more training data is necessary irrespective of topic.

Table 4.13: Cross-topic and topic-specific performances in MCC of 1) Therapist Behaviour Prediction and 2) Client Talk Type Prediction. Three topics are: reducing alcohol consumption (abbr. **Rdc. Drinking**), reducing recidivism, (abbr. **Rdc. Recidivism**) and smoking cessation (abbr. **Rdc. Smoking**). For brevity, we use **boldface** to represent a topic itself (e.g. **Rdc. Drinking**) and *italic* to represent Anno-MI data of dialogues with the topic (e.g. *Rdc. Drinking*). ↓/↑ indicates a decrease/increase from topic-specific to cross-topic performance.

	Using Un Bbalanced Training Data (Original)			Using Balanced Training Data (Augmented)		
Topic	**Rdc. Drinking**	**Rdc. Recidivism**	**Rdc. Smoking**	**Rdc. Drinking**	**Rdc. Recidivism**	**Rdc. Smoking**
Topic-Specific 5-Fold Cross Validation Setup				***Topic-Specific 5-Fold Cross Validation Setup***		
Training Data	80% of AnnoMI, all topics, class unbalanced			80% of AnnoMI, all topics, class balanced		
Test Data	*Rdc. Drinking in 20% of AnnoMI*	*Rdc. Recidivism in 20% of AnnoMI*	*Rdc. Smoking in 20% of AnnoMI*	*Rdc. Drinking in 20% of AnnoMI*	*Rdc. Recidivism in 20% of AnnoMI*	*Rdc. Smoking in 20% of AnnoMI*
Leave-1-Topic-Out Cross-Topic Prediction Setup						
Training Data	AnnoMI w/o Rdc. Drinking	AnnoMI w/o Rdc. Recidivism	AnnoMI w/o Rdc. Smoking	AnnoMI w/o Rdc. Drinking	AnnoMI w/o Rdc. Recidivism	AnnoMI w/o Rdc. Smoking
Test Data	*Rdc. Drinking*	*Rdc. Recidivism*	*Rdc. Smoking*	*Rdc. Drinking*	*Rdc. Recidivism*	*Rdc. Smoking*
Result Format	Leave-1-Topic-Out Result and Averaged Topic-Specific 5-Fold Validation Result					
Therapist Behaviour Prediction Results						
BERT_{adpt}	.78 ↑ (.77)	.67 ↓ (.68)	.73 ↑ (.71)	.77 ↑ (.76)	.65 ↓ (.67)	.72 ↑ (.71)
BERT	.78 ↑ (.76)	.68 (.68)	.73 ↓ (.74)	.77 ↑ (.76)	.68 (.68)	.71 ↑ (.70)
CNN	.57 ↓ (.60)	.52 ↓ (.55)	.58 (.58)	.59 ↓ (.61)	.53 ↓ (.57)	.60 ↑ (.59)
Random Forest	.49 ↓ (.50)	.39 ↓ (.42)	.46 ↓ (.50)	.49 ↓ (.52)	.38 ↓ (.43)	.48 ↓ (.53)
Client Talk Type Prediction Results						
BERT_{adpt}	.34 ↓ (.38)	.19 ↑ (.15)	.35 ↓ (.37)	.29 ↓ (.38)	.12 ↓ (.19)	.33 ↓ (.37)
BERT	.35 ↓ (.38)	.13 ↓ (.19)	.33 ↓ (.37)	.29 ↓ (.35)	.14 ↓ (.15)	.27 ↓ (.35)
CNN	.24 ↓ (.25)	.09 ↓ (.11)	.26 ↓ (.32)	.21 ↓ (.26)	.15 ↑ (.08)	.26 ↑ (.24)
Random Forest	.18 ↓ (.20)	.08 ↓ (.10)	.09 ↓ (.18)	.20 ↓ (.22)	.09 ↑ (.05)	.13 ↓ (.21)

4.9 Discussion

While Anno-MI contains transcripts of MI demonstrations instead of real therapy sessions, we believe that it is the closest approximation possible without privacy violations, while the precise transcription and the accompanying expert annotations further make it more reliable and versatile than similar datasets (e.g. [102]). We note that most of the source videos are from professional therapists and research organisations/institutes dedicated to relevant topics (e.g. reducing substance use), therefore the authenticity of the manifested client-therapist interaction can be considered reliable, as confirmed by the survey responses from the professional annotators. It could also be interesting to explore the domain gap between the corpus and an undisclosed real-world therapy dataset. In particular, as the average duration of the source videos is 7 minutes and thus shorter than usual real-world counselling sessions, in future work we will replicate our experiments on other corpora with longer sessions and then compare the results with those obtained based on Anno-MI. We also note that while client talk type has comparatively lower IAA scores, the performance difference between the trained models and random baselines is substantial, proving the reliability of the annotations on those attributes. As we experimented with attribute prediction based on the current utterance only, the lack of contextualization is also likely to have contributed to the relatively lower performance, which we leave to future work to address. In terms of applications, Anno-MI can be readily used to develop NLP/ML models for MI fidelity, such as generating feedback to help train and supervise counsellors. Example use cases of this nature include 1) categorising current-turn therapist behaviour and/or client talk type, as explored in Sections 4.7 and 4.8, and 2) forecasting next-turn client talk type and/or MI-adherent therapist behaviour. Beyond those natural language understanding settings, Anno-MI can also be used for natural language generation to assist human therapists, such as providing suggestions on what a counsellor could say next, given the past utterances of an ongoing session.

4.10 Conclusion

We introduce Anno-MI [142] a dataset of professionally transcribed and expert-annotated conversations that demonstrate high- and low-quality motivational interviewing. Based on the rich annotations by experienced counsellors, we thoroughly analyse various counselling-related properties at utterance-, dialogue- and corpus-levels. We also create relevant utterance-level prediction tasks and establish strong baseline models. Finally, we examine topic-specific model performance on those tasks and probe the generalisability of the models on new topics. Anno-MI represents a powerful resource for

research in the important direction of counselling-related natural language processing. For future work, we plan to investigate applications of Anno-MI with real-world impact, such as assisting counsellors with real-time session analytics and next-turn suggestions.

Chapter 5

Addressing the challenges of scarce data through augmentation

5.1 Introduction

Recent advancements in NLP captured the interest of the research community in healthcare [67, 79], including mental health and its subdomains such as depression, anxiety, or substance abuse [70]. However, the real-world application of clinical NLP is hampered by multiple elements such as domain complexity, rigorous accuracy and reliability standards and data scarcity [57]. Lastly, recent research highlighted critical concerns on AI fairness [25, 61], which is imperative to address when applying NLP to mental health. As the first step towards addressing these issues, this work adopts data augmentation to improve AI reliability and fairness in the context of scarce mental health data. The work in this chapter leverages our recently released dataset `Anno-MI` [142], consisting of professionally annotated therapy transcriptions in MI [88, 115]. The classification task is modeled to identify therapy quality, one of the `Anno-MI` most unbalanced labels, using each therapist's utterance as input data. In the fairness context, the work inspects therapy topics, e.g., "smoking cessation," "reducing alcohol consumption," or "diabetes management" as the sensitive variable. A detailed quantitative analysis of the effects of data augmentation to balance target and sensitive variables is conducted. The experimental results show little to no impact on CML classifiers but prove that DL ones benefit from augmented data, showing consistent improvement in both accuracy and reliability. Fairness assessment shows that more work on augmentation is required to properly mitigate eventual classification BIAS.

5.2 Material and Methods

Anno-MI[1] [142] contains 110 high-quality and 23 low-quality MI conversational dialogues from a total of 44 topics e.g.: "smoking cessation", "diabetes management", "anxiety management" and others. Therapy quality indicates the therapist's adherence to "general counseling principles taken from the literature on client-centered counseling" [103]. Therapy quality distribution in Anno-MI is heavily skewed towards high-quality (HQ-MI) utterances. This is because the conversations that constitute the dataset belong to MI training videos, which rarely showcase low-quality (LQ-MI) counseling scenarios. To overcome these issues data augmentation techniques are employed.

The works leverages NL-Augmenter[2] [40] to develop an 11-step augmentation pipeline, each one taking one utterance as input. Therefore, for each given utterance, the pipeline generates $n \geq 11$ augmentations (due to certain augmenters potentially producing multiple alternatives for the same utterance). The adopted augmentation techniques include noising, paraphrasing and sampling [72]. Since the augmentation process is unsupervised, caution is taken to avoid using techniques that could lead to semantic changes with respect to the original utterance. With this setup, two augmented versions of Anno-MI are generated, targeting classifier reliability and fairness, respectively.

5.2.1 Problem statement

The work focuses on a binary classification task to detect therapy quality from a single therapist utterance. Each therapist utterance is assigned, to the corresponding conversation quality, in order to formulate the positive and negative examples for the task. Indeed, assessing the quality of MI sessions can boost therapist training and skills assessment, as confirmed from the existing related work on empathy modelling [147, 49, 48, 143], automatic coding of therapeutic utterances [4, 148, 19] and session-level therapist performance [44]. Given the previously mentioned quality skewness, the target variable represents the first potential source of classification unreliability. In this context, Anno-AugMI dataset is generated, which consists of all the therapist utterances from Anno-MI, augmented in order to balance quality proportion. Anno-AugMI creation proceeds in a topic-agnostic fashion, with the goal of obtaining a roughly balanced amount of HQ-MI and LQ-MI utterances across the entire dataset. Since therapy quality is the target of the employed classifiers, this procedure is termed as *target-aware augmentation*. No check is in place with regards to which utterances are augmented, meaning that *target-aware augmentation* merely iterates over the dataset and

[1]Data is available at https://github.com/uccollab/AnnoMI
[2]Code available at https://github.com/GEM-benchmark/NL-Augmenter

augments every low-quality utterance until the target label is balanced. To assess classification fairness, it is necessary to identify the sensitive variable and field-test it with the employed classifiers. For this work the therapy topic (`MI-topic`) is considered as the sensitive variable, as inter-topic fairness guarantees stable performances across a wide range of therapy goals, and because therapy quality in `Anno-MI` is also unbalanced at the topic-level (as shown in Figure 5.1). To address fairness, `Anno-FairMI` dataset is generated, consisting of all the therapist utterances from `Anno-MI`, augmented to balance therapy quality proportion with respect to `MI-topic`. `Anno-FairMI` creation proceeds in a topic-aware fashion, with the goal of having the same amount of `HQ-MI` and `LQ-MI` utterances for each `MI-topic`. Since `MI-topic` is the sensitive variable of our classifier, this procedure is named as *fairness-aware augmentation*. This last procedure introduces the necessity to cut out those `MI-topic` which have no low-quality example since augmentation would have been impossible. As a result, `Anno-MI` and `Anno-AugMI` share all 44 topics (134 conversations), while `Anno-FairMI` keeps only 9 topics (55 conversations), resulting in a much lower pre-augmentation data size. The comparative distribution of topic-wise utterances and average therapy quality per topic is shown in Figure 5.1. The overall distribution of labels in `Anno-MI`, `Anno-AugMI` and `Anno-FairMI` is shown in Table 5.1.

Table 5.1: The overall distribution of high and low-quality therapy utterances.

Dataset	**Total utterances (no.)**	**High quality(%)**	**Low quality(%)**
`Anno-MI`	2601	91%	9%
`Anno-AugMI`	5302	45%	55%
`Anno-FairMI`	9154	50%	50%

5.3 Experiments and Results

A series of experiments are designed for this work, where each experiment's input is based on the output of the preceding ones. The experimental setup is as follows:

- Therapist utterances quality classification of `Anno-MI`.
- Augmentation of `Anno-MI` to balance therapy quality.
- Therapist utterances quality classification of `Anno-AugMI`.
- Fairness assessment of `Anno-AugMI`.
- Augmentation of `Anno-MI` based on `MI-topic`.

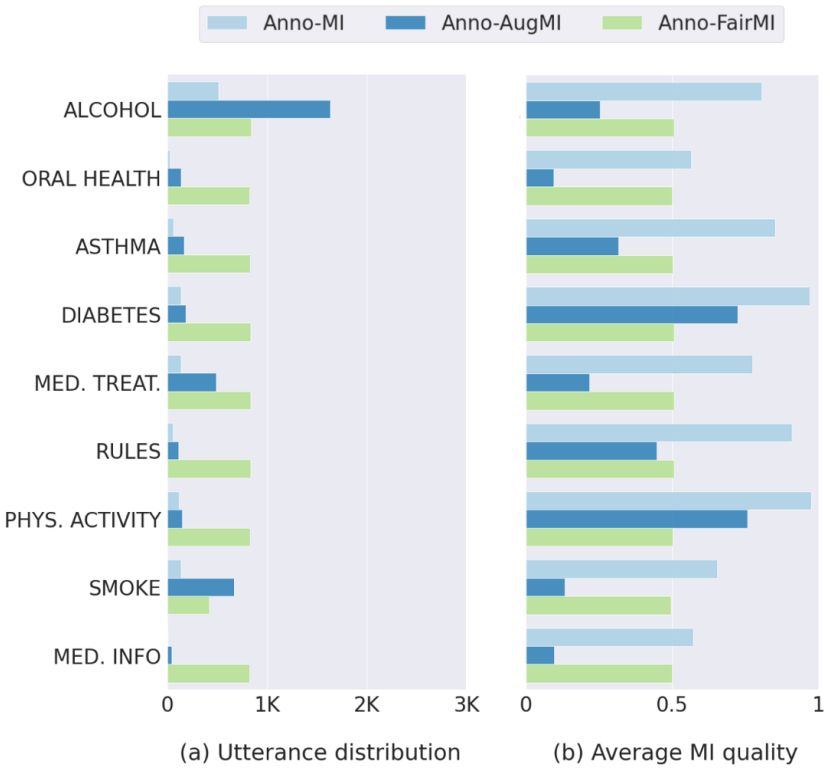


Figure 5.1: Sensitive variable statistics for each dataset. The figure shows topic-wise (a) utterances distribution and (b) average therapy quality. For brevity, only common topics for each dataset are shown.

- Therapist utterances quality classification of `Anno-FairMI`.
- Fairness assessment and BIAS mitigation of `Anno-FairMI`.

SVM and RF as CML classifiers and a BiLSTM-based Dl model with Word2Vec pre-trained word embedding for the embedding layer are employed in this work. Balanced accuracy (BA) and F-1 score are used as performance evaluation metrics of the classifiers. One universal test set is used for all the experiments, created by extracting 400 high-quality and 100 low-quality utterances from `Anno-MI` to avoid any data contamination or bias. The rest of the data is considered as training set and constitutes the basis for the augmentation.

To assess the fairness and mitigate eventual BIAS of the employed classifiers Microsoft FairLearn[3] [8] is used that inspects Selection Rate (SR), False Negative Rate (FNR) and BA as evaluation metrics. Where applicable, "Threshold Optimization" with BA as the target and False Negative Parity as the fairness constraint is adopted. Since `Anno-MI` and `Anno-AugMI` contain multiple topics that lack `LQ-MI` utterances, it is not possible to split training, test and validation data in a way that each partition contains

[3]Code available at https://github.com/fairlearn/fairlearn

both therapy quality classes. The presence of degenerate labels prevents BIAS mitigation, so for these datasets, only the initial metrics values are evaluated.

The classification results of CML and DL approaches for each of the three datasets are summed up in Table 5.3. The obtained results are indicative of consistent low performance of the CML with Anno-MI. The employed augmentation techniques are quite simple so they do not add prominent features to Anno-MI, which can be very helpful in distinguishing classes with bag-of-words representation. This explains the minor performance improvement of the CML algorithms. Since both SVM and RF did not benefit from data augmentation and are comparable to random classifiers, further analysis is not done for CML classifiers. On the other hand, the BiLSTM model shows significant performance enhancement of 23-14% for Anno-AugMI and Anno-FairMI respectively over Anno-MI. Further considerations can be drawn by looking at the confusion matrix in Figure5.2. The initial model, trained on Anno-MI, suffers from the skewed therapy quality distribution and is unable to recognize LQ-MI utterances. This problem also reflects on HQ-MI, with no false positives at all. With *target-aware augmentation* on Anno-AugMI we see more promising results with about 40% of false positives and 14% of false negatives. Finally, with *fairness-aware augmentation* on Anno-FairMI we see pretty much no change in LQ-MI classification, but a considerable drop with HQ-MI, with about 30% false negatives. This can be motivated by the reduced amount of topics in Anno-FairMI, making the BiLSTM suffer from the unseen ones in the test set. In both cases, data augmentation led to an accuracy improvement, which makes our approach promising for future developments [112]. Fairness metrics values for each dataset are shown in Figure 5.3. SR and FNR are apparently ideal for Anno-MI, but this is purely related to the low BA value. Anno-AugMI shows more unbalanced values for SR and FNR, but higher BA than Anno-MI across pretty much every topic. For Anno-FairMI, BIAS mitigation can be run because of the absence of degenerate labels in the training set. Pre-mitigation, Anno-FairMI shows generally more balanced SR, lower FNR, and higher BA than the other two datasets for known topics, and little to no effect after mitigation. However, moving to unseen topics the overall BiLSTM performances greatly worsened, with compromised classification (Figure 5.2) and fairness metrics dropping significantly (Table 5.2).

Table 5.2: The effects of BIAS mitigation on BiLSTM trained on Anno-FairMI. For each metric, the mean value calculated with regard to the sensitive variable (therapy topic) is reported. "TO" stands for "Threshold Optimisation".

Dataset	Selection Rate	False Negative Rate	Bal. Acc.
Anno-FairMI	67.29	23.94	75.72
Anno-FairMI + TO	19.60	72.86	21.89

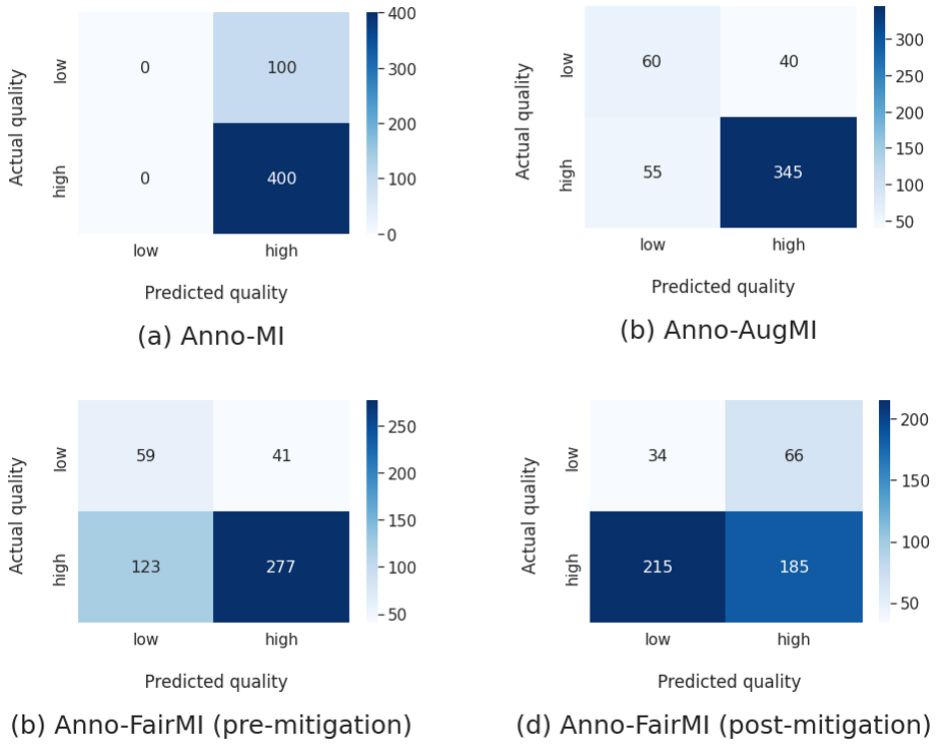


Figure 5.2: Confusion matrix for the BiLSTM trained on each dataset And for Anno-FairMI pre and post-mitigation matrix.

Table 5.3: Performance of CML and DL approaches with Anno-MI, Anno-AugMI, and Anno-FairMI. For each dataset, Balanced Accuracy and F1 score calculated with regards to MI quality are reported.

	SVM		**Random Forest**		**BiLSTM (DNN)**	
Dataset	**Bal.Acc.**	**F-1**	**Bal.Acc.**	**F-1**	**Bal.Acc.**	**F-1**
`Anno-MI`	50.00	44.44	50.75	46.34	50.00	44.44
`Anno-AugMI`	48.87	38.12	50.37	45.78	**73.12**	**71.85**
`Anno-FairMI`	53.87	48.15	51.00	50.99	64.13	59.50

5.4 Conclusion and Future Work

In this work data augmentation is employed to balance target and sensitive variables on the dataset of MI transcriptions `Anno-MI`, resulting in two augmented datasets, namely `Anno-AugMI` and `Anno-FairMI`. The augmentation approaches are evaluated by means of the classification tasks, aimed at recognizing therapy quality. The results show a promising accuracy increase for DL classifiers by using augmented datasets, especially `Anno-AugMI`. This result further motivates to consider other target attributes in future works, such as client talk type or therapist behavior, also extending to other tasks like forecasting. The fairness assessment and BIAS mitigation show that

Figure 5.3: Fairness assessment and BIAS mitigation for BiLSTM on each dataset. For brevity, only common topics for each dataset are shown.

`Anno-FairMI` is too sensitive to unseen topics, opening interesting future work on the adoption of more advanced augmentation techniques. Overall, the outcomes indicate that *target-aware augmentation* is effective at addressing the challenges of unbalanced and scarce data in the mental health domain. Finally, I aim to perform a human evaluation of the developed classifiers, to sanity-check the reliability of the obtained results.

Chapter 6

Conclusion and Future Work

In this thesis, I argue that cultivating a symbiosis between fundamental ML approaches and advanced text representation methods can crucially address the challenges of domain adaptation and reliability of ML in a real-world application. Optimal practices in NLP can alleviate the inherent limits of relevant data to build better NLP models using world knowledge. Each chapter in this thesis poses a novel research question motivated by challenges with (i) unbalanced and small data and (ii) domain adaptation of NLP models. In this process, the thesis makes several contributions, as listed below.

- In Chapter 2, several CML and DL approaches are employed to tackle the multi-classification of clinical records using bag-of-words using TF-IDF and word embedding representation methods coupled with 3 feature selection algorithms. This work extensively investigates the fundamental yet most crucial aspects of data modeling, such as feature selection, data prepossessing, etc., to establish the best practices for NLP tasks. Finally, ensemble models by coupling DL models and CML classifiers mitigate the biased behavior of a single classifier model and improve the single best model's performance prediction stability. The results of this work are promising and assert the efficacy of the employed techniques in dealing with small and imbalanced datasets.
- In chapter 3, K-LM is proposed to use available world knowledge directly in the form of triples to equip LMs with domain knowledge. It is one of the few works available in the literature to tackle domain adaptation. In fact, it can be considered as first work which:
  - introduces Deterministic (context-dependent) and Non-deterministic (context-independent) approaches for knowledge injection in LM.
  - provides a robust pipeline to filter, select and rank the triples for knowledge injection.

  - quantifies the knowledge injection process to mitigate the Knowledge Noise.

  The experimental results have proven the efficacy of K-LM and demonstrate that K-LM is a potential choice for solving knowledge-driven tasks in NLP.

- Chapter 4 introduces Anno-MI, an MI-adherent public dataset comprising both high and low-quality counseling sessions, to address the scarcity of publicly available resources for MI-related NLP research. This dataset is the first of its kind, and the chapter also establishes the baselines for the dataset to be further used by the research community.

- Chapter 5 provides the fairness assessment and BIAS mitigation approaches centered on the classification task modeled from the Anno-MI dataset. It also includes data augmentation techniques useful for the mental health domain, which lead to the creation of two more datasets, Anno-AugMI and Anno-FairMI.

In the context of future work we aim to go past domain adaptation and solve the pertaining NLP problems and bridge knowledge gaps in cross-domain adaptation using world knowledge by means of KGs.

Bibliography

[1] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[2] Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. Realistic re-evaluation of knowledge graph completion methods: An experimental study. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1995–2010, 2020.

[3] KM Annervaz, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati. Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. In *Proceedings of NAACL-HLT*, pages 313–322, 2018.

[4] David C Atkins, Mark Steyvers, Zac E Imel, and Padhraic Smyth. Scaling up the evaluation of psychotherapy: evaluating motivational interviewing fidelity via statistical text classification. *Implementation Science*, 9(1):1–11, 2014.

[5] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[6] Roger Bakeman and Vicenç Quera. Behavioral observation. In *APA handbook of research methods in psychology, Vol 1: Foundations, planning, measures, and psychometrics.*, APA handbooks in psychology®., pages 207–225. American Psychological Association, Washington, DC, US, 2012.

[7] Aakash Bhandari, Vivek Kumar, Pham Thi Thien Huong, and Dang NH Thanh. Sentiment analysis of covid-19 tweets: Leveraging stacked word embedding representation for identifying distinct classes within a sentiment. In *International Conference on Artificial Intelligence and Big Data in Digital Era*, pages 341–352. Springer, 2022.

[8] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A

toolkit for assessing and improving fairness in ai. *Microsoft, Tech. Rep. MSR-TR-2020-32*, 2020.

[9] Robert W Blum, Francisco Inácio Pinkusfeld Monteiro Bastos, Caroline Kabiru, Linh C Le, et al. Adolescent health in the 21st century. *The Lancet*, 2012.

[10] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[11] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[12] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259, 2014.

[13] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[14] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[15] Stefano Bromuri, Damien Zufferey, Jean Hennebert, and Michael Schumacher. Multi-label classification of chronically ill patients with bag of words and supervised dimensionality reduction algorithms. *Journal of biomedical informatics*, 51:165–175, 2014.

[16] Gavin Brown. Ensemble learning. *Encyclopedia of Machine Learning*, 312, 2010.

[17] Paweł Budzianowski and Ivan Vulic. Hello, it's gpt-2-how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. *EMNLP-IJCNLP 2019*, page 15, 2019.

[18] Doğan Can, Panayiotis G Georgiou, David C Atkins, and Shrikanth S Narayanan. A case study: Detecting counselor reflections in psychotherapy for addictions using linguistic features. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[19] Jie Cao, Michael Tanana, Zac Imel, Eric Poitras, David Atkins, and Vivek Srikumar. Observing dialogue in therapy: Categorizing and forecasting behavioral

codes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5599–5611, 2019.

[20] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI conference on artificial intelligence*, 2010.

[21] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder for english. In *Proc. of the 2018 Conference on Empirical Methods in NLP: System Demonstrations*, pages 169–174, 2018.

[22] B Chandra and P Paul Varghese. Fuzzifying gini index based decision trees. *Expert Systems with Applications*, 36(4):8549–8559, 2009.

[23] Zhuohao Chen, Karan Singla, James Gibson, Dogan Can, Zac E Imel, David C Atkins, Panayiotis Georgiou, and Shrikanth Narayanan. Improving the prediction of therapist behaviors in addiction counseling by exploiting class confusions. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6605–6609. IEEE, 2019.

[24] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, Austin, Texas, November 2016. Association for Computational Linguistics.

[25] Alexandra Chouldechova and Aaron Roth. A snapshot of the frontiers of fairness in machine learning. *Communications of the ACM*, 63(5):82–89, 2020.

[26] Domenic V Cicchetti and Sara A Sparrow. Developing criteria for establishing interrater reliability of specific items: applications to assessment of adaptive behavior. *American journal of mental deficiency*, 1981.

[27] William W Cohen. Fast effective rule induction. In *Machine learning proceedings 1995*, pages 115–123. Elsevier, 1995.

[28] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[29] Sergio Consoli, Diego Reforgiato Recupero, and Milan Petkovic, editors. *Data Science for Healthcare - Methodologies and Applications*. Springer, 2019.

[30] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[31] Ofer Dekel and Ohad Shamir. Multiclass-multilabel classification with more classes than examples. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 137–144, 2010.

[32] Danilo Dessì, Rim Helaoui, Vivek Kumar, Diego Reforgiato Recupero, and Daniele Riboni. TF-IDF vs word embeddings for morbidity identification in clinical notes: An initial study. In Sergio Consoli, Diego Reforgiato Recupero, and Daniele Riboni, editors, *Proceedings of the First Workshop on Smart Personal Health Interfaces co-located with 25th International Conference on Intelligent User Interfaces, SmartPhil@IUI 2020, Cagliari, Italy, March 17, 2020*, volume 2596 of *CEUR Workshop Proceedings*, pages 1–12. CEUR-WS.org, 2020.

[33] Danilo Dessi, Rim Helaoui, Vivek Kumar, Diego Reforgiato Recupero, and Daniele Riboni. Tf-idf vs word embeddings for morbidity identification in clinical notes: An initial study. *CEUR Proceddings*, 2596, 2020.

[34] Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain. *Future Generation Computer Systems*, 116:253–264, 2021.

[35] Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, Enrico Motta, and Harald Sack. Ai-kg: an automatically generated knowledge graph of artificial intelligence. In *International Semantic Web Conference*, pages 127–143. Springer, 2020.

[36] Danilo Dessi, Diego Reforgiato Recupero, Gianni Fenu, and Sergio Consoli. Exploiting cognitive computing and frame semantic features for biomedical document clustering. In *SeWeBMeDA@ ESWC*, pages 20–34, 2017.

[37] Danilo Dessì, Diego Reforgiato Recupero, Gianni Fenu, and Sergio Consoli. A recommender system of medical reports leveraging cognitive computing and frame semantics. In *Machine Learning Paradigms*, pages 7–30. Springer, 2019.

[38] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[39] 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, 2019.

[40] Kaustubh D Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahendiran, Simon Mille, Ashish Srivastava, Samson Tan, et al. Nl-augmenter: A framework for task-sensitive natural language augmentation. *arXiv preprint arXiv:2112.02721*, 2021.

[41] A. Dridi and D. Reforgiato Recupero. Leveraging semantics for sentiment polarity detection in social media. *International Journal of Machine Learning and Cybernetics*, 10(8):2045–2055, 2019. cited By 17.

[42] Lisa Ehrlinger and Wolfram Wöß. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48(1-4):2, 2016.

[43] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[44] Nikolaos Flemotomos, Victor R Martinez, Zhuohao Chen, Karan Singla, Victor Ardulov, Raghuveer Peri, Derek D Caperton, James Gibson, Michael J Tanana, Panayiotis Georgiou, et al. Automated evaluation of psychotherapy skills using speech and language technologies. *Behavior Research Methods*, 54(2):690–711, 2022.

[45] Nikolaos Flemotomos, Victor R Martinez, Zhuohao Chen, Karan Singla, Victor Ardulov, Raghuveer Peri, Derek D Caperton, James Gibson, Michael J Tanana, Panayiotis Georgiou, Jake Van Epps, Sarah P Lord, Tad Hirsch, Zac E Imel, David C Atkins, and Shrikanth Narayanan. Automated evaluation of psychotherapy skills using speech and language technologies. *Behavior Research Methods*, 2021.

[46] Christopher A Flores, Rosa L Figueroa, and Jorge E Pezoa. Fregex: A feature extraction method for biomedical text classification using regular expressions. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6085–6088. IEEE, 2019.

[47] James Gibson, David Atkins, Torrey Creed, Zac Imel, Panayiotis Georgiou, and Shrikanth Narayanan. Multi-label multi-task deep learning for behavioral coding. *IEEE Transactions on Affective Computing*, 2019.

[48] James Gibson, Doğan Can, Bo Xiao, Zac E. Imel, David C. Atkins, Panayiotis Georgiou, and Shrikanth S. Narayanan. A Deep Learning Approach to Modeling Empathy in Addiction Counseling. In *Proc. Interspeech 2016*, pages 1447–1451, 2016.

[49] James Gibson, Nikolaos Malandrakis, Francisco Romero, David C Atkins, and Shrikanth S Narayanan. Predicting therapist empathy in motivational interviews using language features inspired by psycholinguistic norms. In *Sixteenth annual conference of the international speech communication association*, 2015.

[50] S Gnanambal, M Thangaraj, VT Meenatchi, and V Gayathri. Classification algorithms with attribute selection: an evaluation study using weka. *International Journal of Advanced Networking and Applications*, 9(6):3640–3644, 2018.

[51] Jun Gu, Wei Feng, Jia Zeng, Hiroshi Mamitsuka, and Shanfeng Zhu. Efficient semisupervised medline document clustering with mesh-semantic and global-content constraints. *IEEE transactions on cybernetics*, 43(4):1265–1276, 2012.

[52] Eren Gultepe, Jeffrey P Green, Hien Nguyen, Jason Adams, Timothy Albertson, and Ilias Tagkopoulos. From vital signs to clinical outcomes for patients with sepsis: a machine learning basis for a clinical decision support system. *Journal of the American Medical Informatics Association*, 21(2):315–325, 2014.

[53] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[54] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[55] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

[56] Shamsul Huda, John Yearwood, Herbert F Jelinek, Mohammad Mehedi Hassan, Giancarlo Fortino, and Michael Buckland. A hybrid feature selection with ensemble classification for imbalanced healthcare data: A case study for brain tumor diagnosis. *IEEE access*, 4:9145–9154, 2016.

[57] Hussein Ibrahim, Xiaoxuan Liu, Nevine Zariffa, Andrew D Morris, and Alastair K Denniston. Health data poverty: an assailable barrier to equitable digital health care. *The Lancet Digital Health*, 3(4):e260–e265, 2021.

[58] Divya Jain and Vijendra Singh. Feature selection and classification systems for chronic disease prediction: A review. *Egyptian Informatics Journal*, 19(3):179–189, 2018.

[59] Nitisha Jain, Jan-Christoph Kalo, Wolf-Tilo Balke, and Ralf Krestel. Do embeddings actually capture knowledge graph semantics? In *European Semantic Web Conference*, pages 143–159. Springer, 2021.

[60] George H John and Pat Langley. Estimating continuous distributions in bayesian classifiers. *In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, page 338–345, 1995.

[61] Jean-Marie John-Mathews, Dominique Cardon, and Christine Balagué. From reality to world. a critical perspective on ai fairness. *Journal of Business Ethics*, pages 1–15, 2022.

[62] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[63] Faiza Khan Khattak, Serena Jeblee, Chloé Pou-Prom, Mohamed Abdalla, Christopher Meaney, and Frank Rudzicz. A survey of word embeddings for clinical text. *Journal of Biomedical Informatics: X*, page 100057, 2019.

[64] Mahnoosh Kholghi, Lance De Vine, Laurianne Sitbon, Guido Zuccon, and Anthony Nguyen. The benefits of word embeddings features for active learning in clinical information extraction. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 25–34, Melbourne, Australia, December 2016.

[65] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[66] Vivek Kumar, Brojo Kishore Mishra, Manuel Mazzara, Dang NH Thanh, and Abhishek Verma. Prediction of malignant and benign breast cancer: A data mining approach in healthcare applications. In *Advances in Data Science and Management*, pages 435–442. Springer, 2020.

[67] Vivek Kumar, Diego Reforgiato Recupero, Daniele Riboni, and Rim Helaoui. Ensembling classical machine learning and deep learning approaches for morbidity identification from clinical notes. *IEEE Access*, 9:7107–7126, 2020.

[68] Vivek Kumar, Abhishek Verma, Namita Mittal, and Sergey V Gromov. Anatomy of preprocessing of big data for monolingual corpora paraphrase extraction: Source language sentence. *Emerging Technologies in Data Mining and Information Security*, 3:495, 2019.

[69] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[70] Aziliz Le Glaz, Yannis Haralambous, Deok-Hee Kim-Dufor, Philippe Lenca, Romain Billot, Taylor C Ryan, Jonathan Marsh, Jordan Devylder, Michel Walter, Sofian Berrouiguet, et al. Machine learning and natural language processing in mental health: Systematic review. *Journal of Medical Internet Research*, 23(5):e15708, 2021.

[71] Robert Leaman, Ritu Khare, and Zhiyong Lu. Challenges in clinical natural language processing for automated disorder normalization. *Journal of biomedical informatics*, 57:28–37, 2015.

[72] Bohan Li, Yutai Hou, and Wanxiang Che. Data augmentation approaches in natural language processing: A survey. *AI Open*, 2022.

[73] Fei Li, Yonghao Jin, Weisong Liu, Bhanu Pratap Singh Rawat, Pengshan Cai, and Hong Yu. Fine-tuning bidirectional encoder representations from transformers (bert)–based models on large-scale electronic health record notes: an empirical study. *JMIR medical informatics*, 7(3):e14830, 2019.

[74] Jundong Li and Huan Liu. Challenges of feature selection for big data analytics. *IEEE Intelligent Systems*, 32(2):9–15, 2017.

[75] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[76] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908, 2020.

[77] Zengjian Liu, Ming Yang, Xiaolong Wang, Qingcai Chen, Buzhou Tang, Zhe Wang, and Hua Xu. Entity recognition from clinical texts via recurrent neural network. *BMC medical informatics and decision making*, 17(2):67, 2017.

[78] Marcus Liwicki, Alex Graves, Santiago Fernàndez, Horst Bunke, and Jürgen Schmidhuber. A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007*, 2007.

[79] Saskia Locke, Anthony Bashall, Sarah Al-Adely, John Moore, Anthony Wilson, and Gareth B Kitchen. Natural language processing in medicine: a review. *Trends in Anaesthesia and Critical Care*, 38:4–9, 2021.

[80] Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

[81] Cynthia Matuszek, Michael Witbrock, John Cabral, and John DeOliveira. An introduction to the syntax and content of cyc. *UMBC Computer Science and Electrical Engineering Department Collection*, 2006.

[82] Stewart Mercer, John Furler, Keith Moffat, Denis Fischbacher-Smith, and Lena Sanci. *Multimorbidity: technical series on safer primary care*. World Health Organization, 2016.

[83] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[84] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2018.

[85] 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.

[86] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[87] William R Miller, Theresa B Moyers, Denise Ernst, and Paul Amrhein. Manual for the motivational interviewing skill code (misc). *Unpublished manuscript. Albuquerque: Center on Alcoholism, Substance Abuse and Addictions, University of New Mexico*, 2003.

[88] William R Miller and Stephen Rollnick. *Motivational interviewing: Helping people change*. Guilford press, 2012.

[89] Theresa B Moyers, Lauren N Rowell, Jennifer K Manuel, Denise Ernst, and Jon M Houck. The motivational interviewing treatment integrity code (miti 4): rationale, preliminary reliability and validity. *Journal of substance abuse treatment*, 65:36–42, 2016.

[90] Marwa Naili, Anja Habacha Chaibi, and Henda Hajjami Ben Ghezala. Comparative study of word embedding methods in topic segmentation. *Procedia computer science*, 112:340–349, 2017.

[91] Tu Dinh Nguyen, Dat Quoc Nguyen, Dinh Phung, et al. A novel embedding model for knowledge base completion based on convolutional neural 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 327–333, 2018.

[92] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[93] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, 2011.

[94] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, pages 271–280, 2012.

[95] Department of Economic and Social Affairs of the United Nations. *World Mortality Report*. United Nations Publications, 2013.

[96] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[97] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[98] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[99] Verónica Pérez-Rosas, Rada Mihalcea, Kenneth Resnicow, Satinder Singh, and Lawrence An. Building a motivational interviewing dataset. In *Proceedings of*

the Third Workshop on Computational Linguistics and Clinical Psychology, pages 42–51, 2016.

[100] Verónica Pérez-Rosas, Rada Mihalcea, Kenneth Resnicow, Satinder Singh, and Lawrence An. Understanding and predicting empathic behavior in counseling therapy. 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 1426–1435. Association for Computational Linguistics, 2017.

[101] Verónica Pérez-Rosas, Rada Mihalcea, Kenneth Resnicow, Satinder Singh, Lawrence An, Kathy J. Goggin, and Delwyn Catley. Predicting counselor behaviors in motivational interviewing encounters. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 1128–1137. Association for Computational Linguistics, 2017.

[102] Verónica Pérez-Rosas, Xinyi Wu, Kenneth Resnicow, and Rada Mihalcea. What makes a good counselor? learning to distinguish between high-quality and low-quality counseling conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 926–935, 2019.

[103] Verónica Pérez-Rosas, Xinyi Wu, Kenneth Resnicow, and Rada Mihalcea. What makes a good counselor? learning to distinguish between high-quality and low-quality counseling conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 926–935, Florence, Italy, July 2019. Association for Computational Linguistics.

[104] Matthew E. 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.

[105] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, 2020.

[106] Boris Pittel. Note on the heights of random recursive trees and random m-ary search trees. *Random Structures & Algorithms*, 5(2):337–347, 1994.

[107] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[108] Kunal Rajput, Girija Chetty, and Rachel Davey. Obesity and co-morbidity detection in clinical text using deep learning and machine learning techniques. In *2018 5th Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*, pages 51–56. IEEE, 2018.

[109] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333, 2011.

[110] Diego Reforgiato Recupero, Mehwish Alam, Davide Buscaldi, Aude Grezka, and Farideh Tavazoee. Frame-based detection of figurative language in tweets [application notes]. *IEEE Comput. Intell. Mag.*, 14(4):77–88, 2019.

[111] D. Reforgiato Recupero and E. Cambria. Eswc'14 challenge on concept-level sentiment analysis. *Communications in Computer and Information Science*, 475:3–20, 2014. cited By 17.

[112] Marnie E Rice and Grant T Harris. Comparing effect sizes in follow-up studies: Roc area, cohen's d, and r. *Law and human behavior*, 29(5):615–620, 2005.

[113] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 74–84, 2013.

[114] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

[115] Stephen Rollnick, William R Miller, and Christopher Butler. *Motivational interviewing in health care: helping patients change behavior*. Guilford Press, 2008.

[116] Andrea Rossi and Antonio Matinata. Knowledge graph embeddings: Are relation-learning models learning relations? In *EDBT/ICDT Workshops*, 2020.

[117] Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You can teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations*, 2019.

[118] Steven L Salzberg. C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993, 1994.

[119] Susan M Sawyer, Sarah Drew, Michele S Yeo, and Maria T Britto. Adolescents with a chronic condition: challenges living, challenges treating. *The Lancet*, 369(9571):1481–1489, 2007.

[120] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

[121] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3060–3067, 2019.

[122] Karan Singla, Zhuohao Chen, David Atkins, and Shrikanth Narayanan. Towards end-2-end learning for predicting behavior codes from spoken utterances in psychotherapy conversations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3797–3803, 2020.

[123] Karan Singla, Zhuohao Chen, Nikolaos Flemotomos, James Gibson, Dogan Can, David C Atkins, and Shrikanth S Narayanan. Using prosodic and lexical information for learning utterance-level behaviors in psychotherapy. In *Interspeech*, pages 3413–3417, 2018.

[124] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007.

[125] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

[126] Michael Tanana, Kevin A Hallgren, Zac E Imel, David C Atkins, and Vivek Srikumar. A comparison of natural language processing methods for automated coding of motivational interviewing. *Journal of substance abuse treatment*, 65:43–50, 2016.

[127] Jiliang Tang, Salem Alelyani, and Huan Liu. Feature selection for classification: A review. *Data classification: Algorithms and applications*, page 37, 2014.

[128] Hans-Christian Thorsen-Meyer, Annelaura B Nielsen, Anna P Nielsen, Benjamin Skov Kaas-Hansen, Palle Toft, Jens Schierbeck, Thomas Strøm, Piotr J Chmura, Marc Heimann, Lars Dybdahl, et al. Dynamic and explainable machine learning prediction of mortality in patients in the intensive care unit: a retrospective study of high-frequency data in electronic patient records. *The Lancet Digital Health*, 2020.

[129] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

[130] Annemarie A Uijen and Eloy H van de Lisdonk. Multimorbidity in primary care: prevalence and trend over the last 20 years. *The European journal of general practice*, 14(sup1):28–32, 2008.

[131] Alper Kursat Uysal and Serkan Gunal. The impact of preprocessing on text classification. *Information Processing & Management*, 50(1):104–112, 2014.

[132] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[133] T Vivekanandan and N Ch Sriman Narayana Iyengar. Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease. *Computers in biology and medicine*, 90:125–136, 2017.

[134] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[135] Samer Abdulateef Waheeb, Naseer Ahmed Khan, Bolin Chen, and Xuequn Shang. Machine learning based sentiment text classification for evaluating treatment quality of discharge summary. *Information*, 11(5):281, 2020.

[136] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[137] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[138] Qiang Wei, Zongcheng Ji, Yuqi Si, Jingcheng Du, Jingqi Wang, Firat Tiryaki, Stephen Wu, Cui Tao, Kirk Roberts, and Hua Xu. Relation extraction from clinical narratives using pre-trained language models. In *AMIA Annual Symposium Proceedings*, volume 2019, page 1236. American Medical Informatics Association, 2019.

[139] Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973*, 2013.

[140] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[141] Chi-Shin Wu, Alex R Luedtke, Ekaterina Sadikova, Hui-Ju Tsai, Shih-Cheng Liao, Chen-Chung Liu, Susan Shur-Fen Gau, Tyler J VanderWeele, and Ronald C Kessler. Development and validation of a machine learning individualized treatment rule in first-episode schizophrenia. *JAMA network open*, 3(2):e1921660–e1921660, 2020.

[142] Zixiu Wu, Simone Balloccu, Vivek Kumar, Rim Helaoui, Ehud Reiter, Diego Reforgiato Recupero, and Daniele Riboni. Anno-mi: A dataset of expert-annotated counselling dialogues. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6177–6181. IEEE, 2022.

[143] Zixiu Wu, Rim Helaoui, Vivek Kumar, Diego Reforgiato Recupero, and Daniele Riboni. Towards detecting need for empathetic response in motivational interviewing. In *Companion Publication of the 2020 International Conference on Multimodal Interaction*, pages 497–502, 2020.

[144] Zixiu Wu, Rim Helaoui, Diego Reforgiato Recupero, and Daniele Riboni. Towards low-resource real-time assessment of empathy in counselling. In *Proceedings of the Seventh Workshop on Computational Linguistics and Clinical Psychology: Improving Access*, pages 204–216, 2021.

[145] Rosemary Wyber, Samuel Vaillancourt, William Perry, Priya Mannava, Temitope Folaranmi, and Leo Anthony Celi. Big data in global health: improving health in low-and middle-income countries. *Bulletin of the World Health Organization*, 93:203–208, 2015.

[146] Bo Xiao, Daniel Bone, Maarten Van Segbroeck, Zac E Imel, David C Atkins, Panayiotis G Georgiou, and Shrikanth S Narayanan. Modeling therapist empathy through prosody in drug addiction counseling. In *Fifteenth annual conference of the international speech communication association*, 2014.

[147] Bo Xiao, Dogan Can, Panayiotis G Georgiou, David Atkins, and Shrikanth S Narayanan. Analyzing the language of therapist empathy in motivational interview based psychotherapy. In *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 1–4. IEEE, 2012.

[148] Bo Xiao, Dogan Can, James Gibson, Zac E Imel, David C Atkins, Panayiotis G Georgiou, and Shrikanth S Narayanan. Behavioral coding of therapist language in addiction counseling using recurrent neural networks. In *Interspeech*, pages 908–912, 2016.

[149] Bo Xiao, Zac E Imel, David C Atkins, Panayiotis G Georgiou, and Shrikanth S Narayanan. Analyzing speech rate entrainment and its relation to therapist empathy in drug addiction counseling. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[150] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[151] Hongteng Xu, Wenlin Wang, Wei Liu, and Lawrence Carin. Distilled wasserstein learning for word embedding and topic modeling. In *Advances in Neural Information Processing Systems*, pages 1716–1725, 2018.

[152] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[153] Liang Yao, Chengsheng Mao, and Yuan Luo. Clinical text classification with rule-based features and knowledge-guided convolutional neural networks. *BMC medical informatics and decision making*, 19(3):71, 2019.

[154] Shanshan Yu, Jindian Su, and Da Luo. Improving bert-based text classification with auxiliary sentence and domain knowledge. *IEEE Access*, 7:176600–176612, 2019.

[155] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

[156] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. Tfidf, lsi and multi-word in information retrieval and text categorization. In *2008 IEEE International Conference on Systems, Man and Cybernetics*, pages 108–113. IEEE, 2008.