

**MAXIMIZING LEARNING EFFICIENCY WITH LIMITED LABELED DATA:  
APPLICATIONS TO HEALTHCARE AND EDUCATION**

---

A Dissertation  
Submitted to  
the Temple University Graduate Board

---

In Partial Fulfillment  
of the Requirements for the Degree  
DOCTOR OF PHILOSOPHY

---

by  
Saman Enayati  
May 2025

Examining Committee Members:

Justin Y. Shi, Advisory Chair, Computer and Information Sciences  
Yan Wang, Computer and Information Sciences  
Stephen MacNeil, Computer and Information Sciences  
Vincenzo Carnevale, External Reader, Biology

©  
Copyright  
2025  
by  
Saman Enayati  
All Rights Reserved

## ABSTRACT

Document classification is essential in domains such as healthcare and education, encompassing three major steps: annotation, training accurate models, and evaluation. Each of these steps is labor-intensive and time-consuming, requiring substantial amounts of labeled data, which is both costly and resource-demanding. This dissertation addresses these challenges by presenting innovative methodologies to enhance annotation efficiency, model training in low-resource settings, and automated scoring.

In the healthcare domain, we tackle the challenges of annotation and training with limited resources. First, we develop a visualization approach for rapid labeling of clinical notes for smoking status extraction. The annotation process is labor-intensive and time-consuming; thus, we introduce a tool that accelerates annotation by clustering similar sentences and highlighting important keywords. This reduces the cognitive load on annotators, resulting in faster and more efficient labeling.

Next, we address the problem of training accurate classifiers in low-resource settings with limited labeled data. In our first approach, MERIT (Minimal Supervision Through Label Augmentation for Biomedical Relation Extraction), we propose using shortest dependency path (SDP) representation and specific distance thresholds to propagate labels and augment high-quality labeled data. This method improves classifier accuracy compared to using limited labeled data alone. We extend this in our second approach by developing an iterative algorithm to learn automatic thresholds for label propagation. This method is tested in various scenarios, including semi-supervised learning, supervised learning, and in-context learning, demonstrating significant improvements in model performance.

In the education domain, we focus on the problem of assessing narratives generated by school-aged children, a task that is both expensive and time-consuming for teachers. We

leverage large language models (LLMs) to learn the scoring patterns of teachers accurately, offering a reliable tool for automated narrative scoring. This approach reduces the subjectivity and resource requirements of manual scoring, providing a scalable and consistent alternative.

Experimental results across these methodologies demonstrate their effectiveness in improving annotation speed, data utilization, and model accuracy. This dissertation contributes to advancing document classification in low-resource settings, offering practical solutions for critical tasks in healthcare and education.

To my beloved family: my mother, Afsaneh, my father, Abedin, and my brother, Sherwin for their endless encouragement and love.

## ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. Justin Shi, for his unwavering support and encouragement throughout the latter stages of my PhD journey. Our collaboration began in Spring 2018 when I served as his teaching assistant, where his expertise, patience, and genuine willingness to teach students left a lasting impact on my approach to academic challenges. Working with Dr. Shi in the final stretch of my PhD has been a privilege, and I always got inspired by our discussions which were invaluable life lessons.

My heartfelt thanks also go to my former advisor, Dr. Slobodan Vucetic, whose mentorship throughout my seven years of working together set me on the path I am on today. His unique perspective on challenges expanded my own, and his emphasis on understanding the depth of the problems has influenced both my academic and personal growth. I am grateful for the foundation he provided, which has shaped my journey.

I would like to thank my current committee members, Dr. Yan Wang, Dr. Stephen MacNeil, and Dr. Vincenzo Carnevale, for their invaluable guidance and support as members of my committee, as well as my former committee members, Dr. Eduard Dragut and Dr. Hongchang Gao, for their early contributions to my academic development. Their insightful feedback greatly enhanced the quality of my dissertation. A special thanks goes to Dr. Yan Wang for his support during latter stages of my PhD. I deeply appreciate his empathy and guidance carried with care and professionalism during my transition to a new advisor. I also extend my gratitude to my collaborator, Dr. Trina Spencer, for her valuable feedback and support throughout my project.

My Ph.D. journey has been deeply enriched by the wonderful friendships I have made inside and outside Temple. These friendships are my treasures and provided me with invaluable emotional support through life's challenges. I am endlessly grateful to have found

such extraordinary friends along this path and I hope to keep these bonds alive, no matter where life takes us. In alphabetical order: Abrar Alrumayh, Arezou Anvari, Ameen Abdel Hai, Amir Azimi, Batool Wazzan, Daniel Saranovic, Emily Thyrum, Erfan Molaei, Farzan Kazemi, Hannah Kim, Jovan Andjelkovic, Jumanah Sameer Alshehri, Mahsa Pashei, Marija Stanojevic, Rafea Aljurbua, Sidra Hanif, Somayeh Keshavarz, and Tanaya Roy.

A special thanks to my amazing friends, Somayeh and Arezou. My immigration journey started with knowing you both, and you've been part of my story since the very beginning. Your presence and support felt like home to me, and I can't thank you enough for how much you helped me through all the difficult times. Thank you for everything!

My heartfelt thanks go to my lab mates, who shared this journey with me and made our lab such a warm and welcoming place with their presence. I will always cherish the time we spent together: Abbey Liu, Ashis Chanda, Beth Garrison, Elmira Talebianaraki, Hanzi Xu, Mathew Kuruvilla, Parsa Esmailkhani, Sai Shi, Sandro Hauri, Shanshan Zhang, Tian Bai, Vahid Mahzoon, Zhuoan Zhou, and Ziyu Yang.

Finally, my deepest appreciation and love go to my family for their endless support. My mother, Afsaneh, without whom I couldn't have made it here—she has always been my source of strength and empowerment. My father, Abedin, my brother, Sherwin, and my partner, Shervin (yes, they share the same name!)—your constant encouragement, love, belief in me, and willingness to listen have been my pillars of strength throughout this journey. I am forever grateful for all you've done for me. I also want to thank my extended family—my grandmother, grandfather, aunts, uncles, and cousins—whose love and support have lifted me up every step of the way.

## TABLE OF CONTENTS

	Page
ABSTRACT . . . . .	iii
DEDICATION . . . . .	v
ACKNOWLEDGEMENTS . . . . .	vi
LIST OF TABLES . . . . .	xii
LIST OF FIGURES . . . . .	xiii
CHAPTER	
1 INTRODUCTION . . . . .	1
1.1 Organization . . . . .	3
2 RELATED WORKS . . . . .	4
2.1 Smoking Status Extraction . . . . .	4
2.2 Low-Resource Relation Extraction . . . . .	5
2.3 Narrative Analysis and Scoring . . . . .	7
3 A VISUALIZATION APPROACH FOR RAPID LABELING OF CLINI- CAL NOTES FOR SMOKING STATUS EXTRACTION . . . . .	9
3.1 Introduction . . . . .	9
3.2 Methodology . . . . .	12
3.2.1 Ordering . . . . .	12
3.2.2 Sentence Visualization . . . . .	13
3.3 Experimental Design . . . . .	14
3.3.1 Results . . . . .	15
3.3.1.1 Round 1 . . . . .	16
3.3.1.2 Round 2 . . . . .	16

	3.3.1.3	Round 3 . . . . .	16
	3.3.2	Ablation Study . . . . .	18
	3.4	Conclusion . . . . .	19
4		MERIT: MINIMAL SUPERVISION THROUGH LABEL AUGMENTATION FOR BIOMEDICAL RELATION EXTRACTION . . . . .	20
	4.1	Introduction . . . . .	20
	4.2	Task Formulation and Background . . . . .	22
	4.3	Methodology . . . . .	23
	4.4	Experiments . . . . .	25
	4.4.1	Datasets . . . . .	25
	4.4.2	Evaluation Metric and Experiment Design . . . . .	26
	4.4.3	Results . . . . .	26
	4.4.4	Ablation Study . . . . .	27
	4.5	Conclusion . . . . .	29
5		LEVERAGING SHORTEST DEPENDENCY PATHS IN LOW-RESOURCE BIOMEDICAL RELATION EXTRACTION . . . . .	31
	5.1	Introduction . . . . .	31
	5.2	Background and Task Formulation . . . . .	34
	5.3	Methodology . . . . .	34
	5.3.1	SDP Representation . . . . .	34
	5.3.2	SDP in Semi-Supervised RE . . . . .	37
	5.4	Experiments . . . . .	39
	5.4.1	Dataset . . . . .	39
	5.4.2	Compared Methods . . . . .	40
	5.4.3	Experimental Setting . . . . .	41
	5.5	Results . . . . .	43

5.5.1	Comparison With Supervised Baselines . . . . .	43
5.5.1.1	Comparison With Non-Encoder Baselines . . . . .	44
5.5.2	Comparison With Semi-Supervised Baselines . . . . .	46
5.5.2.1	Performance on Different Datasets. . . . .	48
5.5.2.2	Performance as a Fraction of Labeled Data Size . . . . .	48
5.5.2.3	Statistical Significance Test. . . . .	48
5.5.2.4	Imputation Bias Analysis. . . . .	49
5.5.2.5	Qualitative Analysis of SSL-RE <sub>sdp</sub> Versus Baselines. . . . .	50
5.5.3	In-Context Learning . . . . .	51
5.5.4	Ablation Study . . . . .	52
5.5.4.1	Choice of Representation On Augmentation Module. . . . .	52
5.5.4.2	Effectiveness of Soft Labels. . . . .	53
5.6	Discussion and Limitation . . . . .	54
5.7	Conclusion . . . . .	55
6	AUTOMATED NARRATIVE SCORING USING LARGE LANGUAGE MODELS . . . . .	56
6.1	Introduction . . . . .	56
6.2	Materials and methods . . . . .	59
6.2.1	Narrative Data Sets . . . . .	59
6.2.1.1	Academic Language of Primary Students (ALPS) Data . . . . .	59
6.2.1.2	Alien Story Data . . . . .	61
6.2.2	Models for Narrative Assessment . . . . .	61
6.2.2.1	GPT3 Fine-tuning . . . . .	62
6.2.2.2	GPT3 In-Context Learning . . . . .	63

6.2.2.3	Baseline Models . . . . .	66
6.2.2.4	Hyperparameter Tuning and Optimization . . . . .	67
6.2.3	Evaluation Metric . . . . .	68
6.2.4	Experimental Design . . . . .	69
6.3	Results . . . . .	70
6.4	Discussion . . . . .	76
6.4.1	Practical Implications . . . . .	78
6.4.2	Limitations and Future Directions . . . . .	78
6.5	Conclusions . . . . .	79
7	CONCLUSION . . . . .	80
	BIBLIOGRAPHY . . . . .	84

## LIST OF TABLES

Table	Page
3.1 The annotation results in Round 1 and 2. . . . .	15
3.2 The results for Round 3. . . . .	15
3.3 Accuracy of ML classifiers on 4 class types. . . . .	17
3.4 Ablation study on the impact of centering and feature visualization. . . . .	18
4.1 Statistics of the dataset used for label augmentation. . . . .	26
5.1 Statistics of each dataset . . . . .	40
5.2 Performance of different RE finetuning architectures when trained using 500 labeled data. . . . .	44
5.3 Comparative analysis of non-encoder based kernel methods using Shortest Dependency Paths (SDP) against our supervised method, which also utilizes SDP for representation. . . . .	46
5.4 The F1 comparison of SSL-RE <sub>sdp</sub> versus SSL baselines. . . . .	47
5.5 T-test analysis of SSL-RE <sub>sdp</sub> versus baselines. . . . .	49
5.6 Using SDP to retrieve NN in few-shot experiments versus random and fixed example selection in prompts in in-context-learning. . . . .	52
5.7 Impact of representation choice in augmentation module, and the resulting performance of RE model. . . . .	53
5.8 Effectiveness of soft label assignment in three datasets using 200 training data. . . . .	53
6.1 Definition of rubrics and their score ranges for each dataset . . . . .	60
6.2 QWK accuracies of different models on 83 test stories from Alien Story Data. . . . .	71
6.3 QWK accuracies of different models on 1,612 ALPS test stories. . . . .	72
6.4 Comparison of finetuning different GPT3 models versus in-context learning with GPT3.5 and GPT4. . . . .	72

## LIST OF FIGURES

Figure	Page
3.1 An illustration of the proposed sequence visualization approach for rapid labeling. . . . .	11
4.1 Label augmentation through local community search. . . . .	23
4.2 Comparison of our approach with RS baseline on three benchmarks biomedical RE datasets. . . . .	27
4.3 The impact of threshold on the final performance. . . . .	28
4.4 Comparison with different feature representations on ChemProt dataset with 200 labeling budgets. . . . .	29
5.1 A dependency parse tree on a biomedical sentence and its shortest dependency path (SDP) tokens (shown in red) between subject (CHEMICAL) and object (GENE) entities. . . . .	33
5.2 Comparison between different representations to fine-tune a RE using a linear layer on top of an encoder. . . . .	35
5.3 Comparing the distribution of imputed labels in the augmented examples (red bars) to their actual labels (blue bars) on DDI, ChemProt, and PPI dataset. . . . .	49
5.4 Qualitative Analysis of SSL-RE <sub>sdp</sub> vs baselines on PPI and ChemProt datasets. LP, ST, DR, RE are denoted as Label Propagation, Self-Training, Dual-RE, and RE-Ensemble. . . . .	50
6.1 GPT3 fine-tuning illustration. . . . .	63
6.2 GPT in-context-learning prompt. . . . .	65
6.3 Data splits for ALPS data. . . . .	70
6.4 IRR comparison of different ML models. . . . .	73

# CHAPTER 1

## INTRODUCTION

Text document classification plays a crucial role in various domains, including healthcare and education, due to its significance in information extraction and analysis. This task involves categorizing text documents into predefined classes or assigning labels to different elements within the documents. The importance of text document classification is evident in applications such as relation extraction in the biomedical domain, smoking status extraction from medical reports, and automating narrative scoring for school-aged children.

In the biomedical domain, relation extraction involves identifying and classifying the relationships between pairs of entities in biomedical text. This task is instrumental in extracting valuable information from scientific articles and clinical records. For example, it enables the identification of drug-disease associations or protein-protein interactions. In healthcare, smoking status extraction from medical reports is a critical task that assists in understanding patient health conditions and tailoring personalized interventions. By accurately classifying smoking status, healthcare providers can better assess patient risk factors and design appropriate treatment plans.

In the education domain, automating the scoring of narrative samples generated by school-aged children is of significant importance. Manual evaluation of these narratives by teachers is time-consuming and subject to human subjectivity. By automating this process, using advanced natural language processing techniques, the assessment of language complexity in different aspects, such as emotion, ending, and character development, can be conducted efficiently and consistently.

However, developing supervised machine learning models for text document classification faces several challenges. The first challenge is the high cost associated with human annotation. The process of manually labeling large volumes of data requires substantial

effort and resources, particularly when domain expertise is required. Moreover, specialized domains often suffer from limited labeled data, making it challenging to train accurate models. The scarcity of labeled data, coupled with the need for expert labeling in certain cases, poses a significant obstacle in developing effective supervised learning models for text document classification.

To address these challenges, this research proposal presents innovative approaches for efficient document classification. Firstly, in the context of smoking status extraction from medical reports, we tackle the issue of human annotation cost by developing a user interface that utilizes visual features. This interface reduces the cognitive load on human annotators, accelerates the annotation process, and enables the collection of a larger volume of labeled data. Consequently, more powerful machine learning models can be trained using this augmented dataset.

Secondly, in the domain of relation extraction from biomedical text, where limited labeled data exists, we propose a label augmentation approach. By carefully propagating labels to unlabeled data points, we can increase the size of the labeled dataset. This augmentation process allows for more comprehensive training and improves the performance of supervised learning models.

Lastly, we aim to automate the narrative scoring process for school-aged children by leveraging GPT, a state-of-the-art language model, through fine-tuning and in-context learning. Our experiments demonstrate the effectiveness of GPT models in low-resource settings, utilizing chain-of-thought prompting with only five examples. Additionally, we showcase fine-tuning in data-rich scenarios to further enhance performance. These approaches collectively enable the development of an automated scoring system that minimizes the need for expert manual evaluation, significantly reducing time and effort while maintaining high accuracy and reliability.

This research proposal is organized as follows. First, we conduct a comprehensive literature review in each sub-domain, highlighting the importance and applications of smoking

status extraction, relation extraction, and narrative analysis. This review sets the foundation for our proposed methods. Subsequently, we delve into the details of each method, addressing the challenges discussed earlier and providing a thorough explanation of the proposed approaches.

Through this research, we aim to contribute to the advancement of text document classification techniques in healthcare and education. By addressing the challenges of limited labeled data and high human annotation costs, we strive to develop efficient and accurate models for information extraction and language analysis.

## **1.1 Organization**

The remainder of this dissertation is organized as follows. Chapter 2 presents a comprehensive literature review. Chapter 3 addresses the challenge of reducing annotation costs in smoking status extraction by introducing a visualization approach for rapid labeling. Chapter 4 introduces MERIT, focusing on the use of Shortest Dependency Path (SDP) representations for efficient label propagation in low-resource biomedical relation extraction. Chapter 5 builds upon MERIT by developing an iterative framework for adaptive threshold learning, enhancing label propagation accuracy, and expanding the application of SDP-based fine-tuning with soft labeling. Chapter 6 explores the capabilities of Large Language Models (LLMs) in narrative assessment, demonstrating their effectiveness in scoring complex narrative elements. Finally, Chapter 7 concludes the dissertation and outlines potential directions for future research.

## CHAPTER 2

### RELATED WORKS

#### 2.1 Smoking Status Extraction

Extracting smoking status of patients from Electronic Health Records [EHR] has been crucial in clinical settings, and especially useful to healthcare providers to select the best care plan for patients at risk of smoking-related diseases. Rajendran & Topaloglu (2020) investigates the application of three Deep Learning models on EHR data to extract the smoking status of patients. Authors compare their approach with traditional machine learning models on both binaries (Smoker vs Non-Smoker) and multi-class classification (Current Smoker vs. Former Smoker vs. Non-smoker) tasks. Wang et al. (2016) extracts smoking status from three different sources such as narrative texts, patient-provided information, and diagnosis codes. They conclude that narrative text proves to be the most useful source for smoking status extraction. Palmer et al. (2019); Hegde et al. (2018) develop rule-based algorithms to determine tobacco use by patients. Palmer et al. (2019) further identify the cessation date and smoking intensity of patients. Common for the aforementioned work on smoking status extraction is a need to label sentences and train an appropriate machine learning model. None of those papers discuss issues related to labeling nor attempt to reduce labeling costs.

A common approach to annotate a large amount of data is through crowdsourcing Fang et al. (2014); Good & Su (2013); Lim et al. (2020). It has been used in variety of tasks such as Image Classification Fang et al. (2014), Bioinformatics Good & Su (2013), and Text mining Li et al. (2020). Although crowdsourcing is a cost-effective way to collect labeled data, it can still be costly when the required labeling effort is significant. Moreover, when using imperfect annotators with varying levels of expertise, it is important to develop appropriate label integration approaches Settles (2011). Beyond the crowdsourcing issues,

one popular approach to reduce labeling costs is to apply Active Learning and label only the most informative examples Fang et al. (2014).

More recently, Human-In-the-Loop [HIL] approaches were proposed to improve the efficiency of annotation Klie et al. (2020); Kim & Pardo (2018). Kim & Pardo (2018) present a HIL system for sound event detection, which directs the annotator’s attention to the most promising regions of an audio clip for labeling. Klie et al. (2020) apply a similar technique on Entity Linking [EL] task, in which the machine learning component makes recommendations about the most relevant entries in a knowledge base, and the annotator selects the correct candidate. The recommender improves itself based on the obtained feedback. In addition, Qian et al. (2020) present an interface for entity normalization annotation in which they measure the number of clicks in a tool to quantify the human effort.

While many papers attempt to minimize labeling effort, a vast majority of them are measuring the effort by counting the number of labeled examples. There are very few papers Zhang et al. (2019) that measure labeling effort in terms of elapsed time. The uniqueness of our work is in demonstrating that annotation speed can be significantly impacted by the way data is presented to an annotator. Furthermore, our work is specific in its focus on an extreme labeling scenario where the task is to label the complete corpus in order to maximize the prediction accuracy.

## **2.2 Low-Resource Relation Extraction**

There are several major approaches to dealing with small labeled datasets in low-resource relation extraction (RE). These approaches include active learning, weak supervision Mintz et al. (2009), and semi-supervised learning Ouali et al. (2020). The main assumption in all those approaches is large quantities of unlabeled data exists. Active Learning aims to minimize the cost of collecting labeled data by carefully selecting the

most informative samples from unlabeled data to be labeled by human annotators Hanneke (2009); Ouali et al. (2020).

In weak supervision, techniques such as heuristics or rules are used to generate noisy labels on unlabeled data. The goal is to train a model that can generalize to new examples. Mintz et al. (2009) proposed a distant supervision method where two entities co-occurring in a sentence are labeled with their relations from a knowledge base, regardless of their context. Qu et al. (2018) and Zhou et al. (2020) generated labeling rules from the tokens between entity pairs using different strategies. Qu et al. (2018) used the tokens on the shortest dependency path of an entity, while Zhou et al. (2020) used sequences of tokens (frequent phrases) between entity pairs. Ratner et al. (2020) introduced an approach where experts write labeling rules and a model is trained to resolve disagreements among the rules, reducing labeling noise.

Semi-supervised learning (SSL) is another approach that utilizes both a small amount of labeled data and a larger amount of unlabeled data Ouali et al. (2020). SSL methods, such as self-training Rosenberg et al. (2005) and graph-based approaches Zhu & Ghahramani (2002); Chen et al. (2006), leverage the unlabeled data to refine the model’s understanding of the data and improve its performance.

Label propagation (LP) is a popular graph-based SSL method that propagates labels from labeled nodes to unlabeled nodes based on their similarity Zhu & Ghahramani (2002); Chen et al. (2006). LP does not depend on a predictor and uses an adjacency matrix to represent the relationships between nodes. The nodes correspond to the data points, and an  $n \times n$  adjacency matrix  $T$  represents the weight between any two nodes, which is calculated based on any distance metric. Let  $Y$  be a  $n \times C$  corresponds to the scores for each node, where  $C$  is the probability distribution over classes. The algorithm converges to an optimal solution by iteratively computing  $Y \leftarrow TY$ . Chen et al. (2006) applied LP to semi-supervised RE using lexical and syntactic features between entity pairs. However, the

choice of features and dynamic feature transformation in LP algorithms remains an open research question.

In contrast, predictor-based SSL approaches such as self-training and dual training utilize the predictions of a predictor to provide weak supervision Ouali et al. (2020). Self-training expands weak labels iteratively based on the predictor’s predictions, while dual training incorporates a retrieval model to retrieve additional pseudo-labeled data for a given relation label Lin et al. (2019).

These approaches introduce labeling noise and still require expert knowledge or manual annotation. Furthermore, the quality of imputed labels in SSL methods can be low when the labeled dataset is small. Additionally, the choice of distance metrics and neighborhood thresholds in graph-based SSL methods affects their performance.

To address the challenges mentioned above, we propose leveraging the labeled data itself and employing an appropriate distance-based representation to measure the similarity between labeled and unlabeled data points.

### **2.3 Narrative Analysis and Scoring**

Given the value of narratives, several researchers have investigated methods to increase the efficiency of scoring narratives Silva (1999). Some researchers applied rule-based measurements Hsu & Thompson (2018) and human evaluation Khalpada & Garg (2021) to evaluate generated narratives. In oral narrative field, researchers investigated the validity of children’s story retell and narrative comprehension assessment Gillon et al. (2023) for educational benefits.

Inspired by the technical knowledge Ramesh & Sanampudi (2022); Susanti et al. (2023) in Automated essay scoring (AES), researchers began to analyze narratives using machine learning and deep learning methods Ranade et al. (2022). Automated narrative analysis tends to evaluate the quality of narrations from different rubrics Fox et al. (2022). Sim-

ilar to AES, most automated narrative analysis can be finalized as a scoring task, which could be either regression or classification task. Researchers used random forest to train a supervised classifier to predict the quality of narratives Somasundaran et al. (2015). Convolutional Neural Networks (CNNs) LeCun et al. (2015) and Gated Recurrent Units (GRUs) Dey & Salem (2017) have been implemented to score essays Tashu et al. (2022) as a regression task. Moreover, a transformer-based model, BERT Devlin et al. (2018), was used to improve the performance of scoring narrations Jones et al. (2019); Fernandez et al. (2022) based on Test of Narrative Language (TNL) Gillam & Pearson (2004b) and Monitoring Indicators of Scholarly Language (MISL) Gillam et al. (2017), which are standard assessment tools for evaluating and tracking macro-structure features of narratives.

The recent development of large language models (LLMs), such like GPT-3 and Instruct-GPT Brown et al. (2020); Ouyang et al. (2022), have shown promising results across numerous natural language processing (NLP) tasks. Multiple investigations have explored the issue of contextual biases in generative models, such as GPT-3, during narrative formation Lucy & Bamman (2021). In addition, researchers have employed GPT-3 for essay grading under diverse evaluation criteria by supplying instructional prompts to the model Mizumoto & Eguchi (2023a). Despite these achievements, no studies have yet focused on applying LLMs for automated narrative analysis.

## CHAPTER 3

### A VISUALIZATION APPROACH FOR RAPID LABELING OF CLINICAL NOTES FOR SMOKING STATUS EXTRACTION

#### 3.1 Introduction

Deep learning algorithms achieve state-of-the-art accuracy on a range of natural language processing tasks. However, to achieve high accuracy, deep learning algorithms typically require a lot of labeled data. In extremely error-sensitive applications, such as those in the medical domain, the trade-off between labeling effort and prediction accuracy is strongly skewed towards maximizing the accuracy. In such applications, data labeling arises as the most costly and human-intensive step during the development of deep learning models. In this paper, we focus on a scenario where the requirement is to label all available data because the goal is to maximize the accuracy using the available corpus of documents. In such a scenario, none of the labeling shortcuts developed in the machine learning community such as active learning are of much help on their own.

Our focus is on presenting textual information to human annotators in a way that minimizes their cognitive load, thus improving their focus, and maximizes their labeling speed, thus reducing the cost of labeling. Our proposed visualization approach is fine-tuned to enable text labeling in the specific application where the objective is to extract information about smoking status of patients from their medical notes. Smoking status of patients is critical information in many practical applications, ranging from recruiting participants in clinical trials to determining medical and life insurance premiums for prospective customers.

Smoking status extraction is a specific instance of information extraction problems. Our visualization approach relies on several key observations about this particular type of problem. We first observed that smoking status could typically be extracted from sentences that

contain one of the smoking keywords such as smoke, smoking, tobacco, nicotine. Thus, our first step was to extract from the corpus only sentences containing one of those keywords. Our second observation was that smoking status can typically be deduced from several words surrounding the keyword. Thus, it might be possible to prune very long sentences to sub-sentences surrounding the keyword without loss of information. This observation allows reserving only a single line to display each relevant sentence.

Our third observation is that the space of possible smoking-related sentences occurring in clinical notes is relatively limited and that for any smoking-related sentence there are likely very similar sentences in the corpus. We hypothesized that displaying similar sentences next to each other would allow human annotators to process the text much faster than if sentences are shown in random order. Our fourth observation is that some common discriminative keywords reveal the smoking status, such as denies, quit, former, packs. We hypothesized that highlighting those keywords in the text could allow a human annotator to work faster.

Our final observation was that by training a predictive model on the currently available labels, even when the number of available labels is relatively small, would likely result in prediction accuracy that is significantly higher than a baseline that assigns labels randomly or based on the majority class labels. Thus, providing labels obtained by the current prediction model would allow a human annotator to skip the correctly labeled sentences and only enter the labels for the incorrectly labeled ones. As the number of labels increases, the prediction model's accuracy is expected to improve, reducing the effort required to correct labels and thereby enhancing labeling speed.

The resulting visualization approach developed by exploiting the stated observations is illustrated in Figure 3.1. A panel at the top shows 7 randomly selected smoking-related sentences from our corpus. A panel at the bottom shows the same sentences displayed using our approach. The main features of our visualization approach are (1) sentence ordering, (2) sentence centering around the smoking keyword, (3) text annotation to emphasize dis-

criminative keywords, and (4) displaying of the predicted labels. We are claiming, and our user study (described in Section 3.3) confirms it, that the bottom panel makes it much easier and faster for a human annotator to label a large corpus of smoking related sentences for the smoking status of a patient.

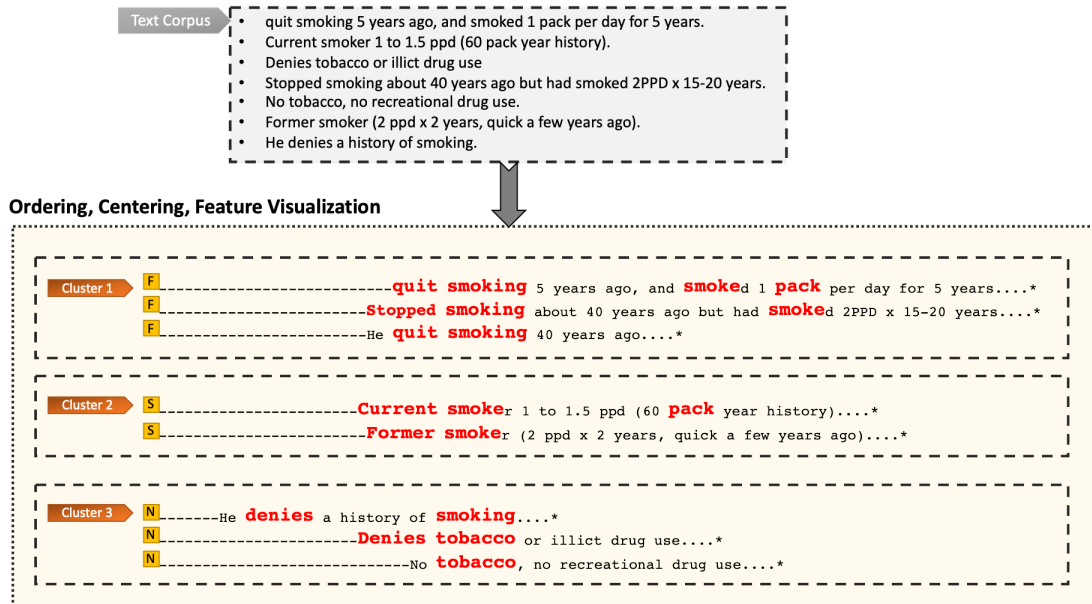


Figure 3.1. An illustration of the proposed sequence visualization approach for rapid labeling. The predicted labels for each sentence are shown inside the yellow boxes. where N refers to Non-Smoker, F to Former Smoker, and S to Smoker. Only the 5th sentence in the bottom panel is misclassified by the current prediction model and has to be overwritten by a human annotator.

To produce the bottom panel in Figure 3.1, we had to decide (1) what are the smoking keywords, (2) what keywords are discriminative of the smoking status, (3) how to order the sentences, (4) how to provide predicted labels, (5) what to do during the cold start when no or very few sentences are labeled, and (6) how to implement the visualization approach. Details about the proposed approach are provided in Section 3.2. In Section 3.3 we describe the experimental design, explain our user study, and provide experimental results that convincingly indicate the usefulness of the proposed approach.

## 3.2 Methodology

**Problem Definition:** Given a document corpus  $D$  representing clinical notes of patients from which a set of  $N$  unlabeled smoking-related sentences  $S_1, S_2, \dots, S_N$  is extracted, the goal is to ask human annotators to label all  $N$  sentences for smoking status. There are 4 types of labels: Smoker (S), Non-Smoker (N), Former Smoker (F), and Other (O), where Others refer to sentences that do not reveal the smoking status.

In this section, we describe a visualization approach that improves human annotation speed. The main components of the approach are sequence ordering, label prediction, and text visualization. The details are explained in the following subsections.

### 3.2.1 Ordering

Our goal is to order sentences in a computationally-efficient manner by combining clustering and alignment algorithms. We use clustering to find groups of similar sequences that will subsequently be ordered with help of an alignment algorithm.

In order to cluster sentences, we rely on their vector embeddings. In particular, we use sequence embeddings of the pre-trained BERT model Devlin et al. (2019). K-Means Clustering, whose computational cost is  $O(N)$  as implemented by Pedregosa et al. (2011), is used to find  $k$  clusters, where  $k$  is selected such that the average cluster size is limited to a specified size.

Sentences in each cluster are then ordered, such that neighboring sentences are perceived by a human annotator to be as similar as possible. Rather than ordering sentences based on BERT embeddings, we instead resort to sequence alignment distance, which we hypothesize are closer to human perception of similarity. In particular, we apply Needleman–Wunsch algorithm [NWA] <sup>1</sup> Needleman & Wunsch (1970), which is a dynamic programming algorithm that finds a similarity score between a pair of sentences in  $O(L^2)$  time,

---

<sup>1</sup> [http://emboss.sourceforge.net/docs/emboss\\_tutorial/node3.html](http://emboss.sourceforge.net/docs/emboss_tutorial/node3.html)

where  $L$  is the length of a sentence, For each cluster, we create a pairwise score matrix,  $Score$ , of size  $N_c \times N_c$ , where  $N_c$  is the number of sequences within the cluster  $c$ .

To find the order of the sentences in each cluster, we apply the following greedy algorithm. It starts by selecting the first sentence at random. The next sentence is its nearest neighbor, according to  $Score$  matrix. The process continues by adding the nearest neighbors of previous sentences.

### 3.2.2 Sentence Visualization

Once the sentences are sorted, our next objective is to display them in a way that reduces the cognitive load of a human annotator. Our first idea is to center the sequences around smoking-related keywords such as Smoke, Smoking, Tobacco, Nicotine. We find those keywords by applying word2vec Mikolov et al. (2013) to our document corpus  $D$  and by finding neighbors of word Smoke in the resulting embedding. Then, we manually select neighbors that are indicative of smoking-related sentences.

According to the maximum screen width, we align the sentences such that the smoking keyword appears in the middle of the screen. In addition, we fill the empty spaces before the sentence starts with dashes (-) to improve readability.

Our labeling approach proceeds in batches. After selecting the first batch of  $M$  unlabeled sentences at random (in our experiments we use  $M = 200$ ), we do not display any predicted labels and orders. After we obtain labels for the first batch, we train a baseline machine learning model such as logistic regression using the bag of words representation (in our experiments we used the most frequent 500 non-stop words). Then, we analyze the statistical significance of the logistic regression weights and select  $K$  words associated with the most significant weights as discriminative words such as cigarette, denies, quit, former, packs.

We select the second batch of unlabeled sentences at random, order them, and display them centered with the discriminative words in bold red font to improve readability. In addition, we display the predicted labels by the logistic regression next to the ordered sentences.

Rather than building a specialized sentence visualization and annotation tool, we use MS Excel<sup>2</sup>. Each sentence occupies one row in the Excel spreadsheet, where the first column is reserved for prediction labels, and the second column is reserved for the centered annotated sentences. An advantage of Excel is that it enables the use of the built-in cell drag feature to quickly change annotations of neighboring sentences. In addition, we use Courier as the font format, since it is a monospaced font type. The monospaced font displays each character or letter in the same amount of horizontal space. As a result, it makes the alignment and centering precise.

We continue selecting batches, labeling them, and retraining the prediction models. Once the number of labels becomes sufficiently large (1,000 in our experiments) we replace logistic regression with deep learning. We also allow for the batches to become larger over time.

### **3.3 Experimental Design**

We performed our experiments using 52,726 discharge notes from the MIMIC-III dataset Johnson et al. (2016), which contains de-identified records of the Beth Israel Deaconess Medical Center’s Intensive Unit emergency department patients from 2001 to 2012.

We defined smoking-related keywords by selecting keyword smoke and its selected word2vec nearest neighbors. We collected 26 unique keywords. Using those keywords, we found 34,149 unique matching sentences.

---

<sup>2</sup> <https://www.microsoft.com/en-us/microsoft-365/excel>

Table 3.1. The annotation results in Round 1 and 2. The experiments are conducted in the same order as the numbers indicate. Each group contains 200 sentences. Unordered refers to the baseline, and Ordered is our visualization approach.

<b>Groups &amp; Settings</b>	<b>User 1 (mins)</b>	<b>User 2 (mins)</b>	<b>Rate User 1 (Sent/min)</b>	<b>Rate User 2 (Sent/min)</b>	<b>Total rate (Sent/min)</b>
Round 1					
<b>Batch1 (Unordered)</b>	27	19	7	10	17
Round 2					
<b>Batch1 (Unordered)</b>	19	17	10	11	21
<b>Batch2 (Ordered)</b>	12	11	16	17	33
<b>Batch3 (Ordered)</b>	<b>11</b>	<b>9</b>	<b>17</b>	<b>21</b>	<b>38</b>
<b>Batch4 (Unordered)</b>	16	16	12	12	24

Table 3.2. The results for Round 3. The experiments are conducted in the same order as the numbers indicate. Each Group contains 500 samples. The labels for these experiments are provided by fine-tuned Clinical BERT model. Unordered refers to the baseline, and Ordered is our visualization approach.

<b>Groups and Settings</b>	<b>User 1 (mins)</b>	<b>User 2 (mins)</b>	<b>Rate User 1 (Sent/min)</b>	<b>Rate User 2 (Sent/min)</b>	<b>Total rate (Sent/min)</b>
<b>Batch1 (Unordered)</b>	40	35	12	14	26
<b>Batch2 (Ordered)</b>	23	23	21	21	42
<b>Batch3 (Ordered)</b>	<b>19</b>	<b>20</b>	<b>26</b>	<b>24</b>	<b>50</b>
<b>Batch4 (Unordered)</b>	34	34	14	14	28

### 3.3.1 Results

We evaluate the effectiveness of our proposed approach in three different rounds of labeling. We performed a user study with 2 human annotators (the first two co-authors of this paper) to measure labeling time in each of the 3 rounds of labeling. The total number of sentences annotated by each user in our experiments was 3,000 sentences each. In addition, in Section 3.3.2, we performed an ablation study to analyze the impact of different components of the proposed visualization approach.

In addition to labeling time, we also report the labeling rate, which is the number of sentences labeled per minute:

$$Rate = \frac{\# \text{ of annotated sequences}}{\text{elapsed time}}$$

In the following subsections, we explain the basics of each baseline method as well as the experimental design for each round of labeling.

**3.3.1.1 Round 1** In this round of the experiment, we select 200 random sentences. We display them in the same way as it is shown in the upper panel in Figure 3.1. Once we obtain the labels from the first batch, we train a logistic regression model. The first row of Table 3.1 shows the annotation details.

**3.3.1.2 Round 2** We asked users to annotate 800 sentences in 4 batches. We chose the Latin square design to proceed as unordered, ordered, ordered, and unordered batches. We have also use logistic regression model to predict the labels for all the batches. Table 3.1 demonstrates the result of this round.

On average, the annotation rate using our method is  $1.9\times$  compared to round 1. Additionally, it is  $1.5\times$  faster compared to the unordered set in Round 2. By repeating the annotation task in batches 3 and 4, we can speed up the rate in our method by 15% (from 33 to 38) and in the unordered set by 14% (from 21 to 24).

**3.3.1.3 Round 3** We annotated 2,000 sentences in 4 batches, each batch containing 500 sentences. Similar to Round 2, we set up the experiments with the Latin Triangle mixture design (unordered, ordered, ordered, unordered).

Given the annotated data from Round 1 and 2, we replaced the classifier with a deep learning algorithm. We use the Clinical BERT, which is pretrained on all the discharge summary notes in the MIMIC dataset. We split the data into 800 training and 200 for

testing. The hyperparameters are selected according to Devlin et al. (2019). We set the batch size to 16, learning rate to  $2e-5$ , maximum sentence length to 200, and fine-tuned it for 4 epochs. We also experimented with SVM and logistic regression, as shown in Table 3.3.

According to Table 3.2, the annotation rate increased from Round 2 to Round 3 by 29% (from 35.5 to 46) with our approach. However, it increased by 16% (from 22.5 in Round 2 to 27 in Round 3) using the baseline approach.

Comparing the annotation speed in Round 3, our approach is  $1.7\times$  faster than the baseline (46 compared to 27). Since the size of the batches increased in Round 3, there was more redundancy in the sentences and our approach was more helpful to the annotators than in Round 2. In particular, ordering resulted in smoother transitions between sentences, which contributed to faster human annotation.

Last but not the least, by repeating the labeling task, we expect users to get used to the data, and therefore, we expected the annotation rate to increase regardless of the visualization approach. Confirming this assumption, users on average got 19% faster with our method during Round 3 (rate increased from 42 to 50), while they got only 7% faster with the baseline approach (rate increased from 26 to 28).

Table 3.3. Accuracy of ML classifiers on 4 class types. All the classifiers are trained to predict 4 classes: Smoker, NonSmoker, Former, and Other. Baseline accuracy is the fraction of the majority class in the test set. In Round 1, there are 800 training and 200 test sentences. In Round 2, there are 3,400 training and 600 test sentences.

<b>Model</b>	<b>Accuracy Round 1</b>	<b>Accuracy Round 2</b>
Baseline	0.35	0.36
Logistic Regression	0.76	0.79
SVM	0.78	0.80
Fine-tuned Clinical BERT	0.78	0.89

### 3.3.2 Ablation Study

In this section, we analyze the impact of two components of our system on the final annotation rate. We asked one of the users to annotate an additional 1,000 sentences. We split the set into two groups, each group with 500 samples. First, we studied the impact of centering. Therefore, we aligned all the data to the left and kept the ordering and feature visualization. Second, we removed the feature visualization component, and kept the ordering and centering. Table 3.4 shows the results of these two experiments.

Table 3.4. Ablation study on the impact of centering and feature visualization. In the first row, we do not center the sentences around the smoke keywords. In the second row, we do not highlight the important features.

<b>Components</b>	<b>User 2 (mins)</b>	<b>Rate User 2 (Sent/min)</b>
<b>No centering</b>	22	22
<b>No coloring</b>	21	23

According to the results for Round 2 in Table 3.2, the highest rate for User 2 was 24 sentences per minute. However, when we removed the centering component, the rate decreased by 8%, to 22 per minute. In addition, by removing the coloring component, the rate decreased by 4%, to 23 per minute. The centering component had a stronger impact on the labeling rate than the coloring component. However, both of the removals reduced the rate of labeling.

Given the annotated data from the ablation study, and adding all the labeled data from the first and second rounds, we re-trained all the classifiers on 3,400 training sentences and used 600 sentences for testing. We observed 15% improvement in the BERT model accuracy and 3% improvement in the Logistic Regression model accuracy compared to the models trained on Round data.

### 3.4 Conclusion

We presented a visualization approach that enables rapid annotation of sentences for smoking status of patients. Our framework contains three main components: sentence ordering, sentence presentation, and sentence labeling by the prediction model. Our approach does not depend on high-quality ML predictors to provide initial labels. The display has a significant impact on speeding up the annotation process. We evaluated our visualization approach with a user study on sentences from MIMIC-III discharge summaries. We achieved close to  $3\times$  faster annotation rate compared to the baseline method that displayed sentences randomly in their original shape. As the annotation progressed, as the batches of unlabeled sentences became larger, and as the prediction models improved, the annotation speed kept increasing in our user experiments. The proposed visualization approach is applicable to similar text classification tasks. It is a topic of further research to study how to modify the presented approach to make it applicable to a large number of text annotation tasks in natural language processing.

## CHAPTER 4

### MERIT: MINIMAL SUPERVISION THROUGH LABEL AUGMENTATION FOR BIOMEDICAL RELATION EXTRACTION

#### 4.1 Introduction

Relation Extraction (RE) is defined as classifying a type of relationship between a pair of entities occurring in a text passage. For example, given the sentence “Ciprofloxacin has some effect on Pyelonephritis”, we can infer the relation “May Treat” between two entities Ciprofloxacin and Pyelonephritis, and form a triplet (subject, relation, object). The extracted triplets from the text can be used for knowledge base population, question answering, or information retrieval. Recent advances in Deep Learning allow training very accurate Hassantabar et al. (2021) models Zhou et al. (2021); Lee et al. (2020); Gupta et al. (2019); Wei et al. (2019); Malekzadeh et al. (2021) when a large amount of human-annotated training data is available. However, collecting training data is a costly and human-intensive process. In some specialized domains such as biomedical text, human annotation is particularly challenging and costly because it can only be done by biomedical experts. Therefore, RE training data in the biomedical domain can often be very small and result in RE models with low accuracy.

Weak or distant labeling is a popular approach for addressing label scarcity issues in many applications, including RE Krasakis et al. (2018); Boudjellal et al. (2020). Distant supervision is applied to heuristically align entities to a given knowledge base (KB) with little annotation effort. However, distant supervision requires KB existence and entity co-occurrence without taking the underlying context between two entities into account. In weak labeling, labeling rules are used by string matching to automatically provide labeled-data Qu et al. (2018); Zhou et al. (2020); Krasakis et al. (2018); Liu (2018); Ratner et al. (2016, 2020), which are often more efficient than using a KB for distant supervision. How-

ever, exact string matching limits the generalizability of the rules, and thereafter causes low coverage of data and labeling noise. To tackle the labeling noise issue, data programming Ratner et al. (2016, 2020) aims to annotate the corpus by fitting a model to resolve any disagreements among the rules. Approaches to address the coverage of the labeling rules include differentiable soft-assignment of the rules to an unlabeled portion of the corpus Zhou et al. (2020); Ren et al. (2020); Meng et al. (2018). Although these approaches provide higher coverage of the rules, they still suffer from labeling noise. Moreover, generating labeling rules is a costly and inexact process and depends on the skill of a user to convert their knowledge into useful rules. There have been also efforts in reducing annotation costs in Biomedical domain by proposing a visual interface to accelerate annotation Enayati et al. (2021) or algorithms to generate semi-structured annotations Katic et al. (2021). However, these solutions are less applicable to RE domain.

As an alternative to distant and weak labeling, we propose a simple and efficient approach, MERIT, to automatically increase the number of labels given a small labeled data set. Our main observation is that the nearest neighbors of a sentence representing a particular relation between its entities are likely to Zhu & Ghahramani (2002) represent the same relation. Thus, given a labeled sentence, we transfer its label to all its neighboring sentences. An open question to be studied in this paper is what is a good definition of the neighborhood in the RE task, which requires us to answer what is an appropriate distance measure and what is an appropriate distance threshold. The benefit of the MERIT is that it does not require any expert knowledge and that it is very easy to implement and computationally inexpensive. The proposed approach can be combined with other approaches dealing with data labeling scarcity such as weak labeling and active learning, including uncertainty-based sampling Seung et al. (1992); Wang et al. (2013, 2015) and clustering-based sampling Nguyen & Smeulders (2004); Wang et al. (2017).

Despite the efforts that have been made in synthetic training data augmentation Papanikolaou & Pierleoni (2020); Hassantabar et al. (2019) and other types of label augmen-

tation Solmaz et al. (2022), MERIT exploits the unlabeled portion of corpus to augment the limited available hand-labeled data with high-quality weak labels, which results in a more accurate RE model. We perform extensive experiments on three benchmark biomedical relation extraction datasets.

## 4.2 Task Formulation and Background

Given a sentence  $s_i$  and an entity pair  $(e_{\text{subj}}, e_{\text{obj}})$ , the task can be transformed to a classification problem. A classifier can be built to map the relation between subject and object entities into predefined relation types  $\{R \cup NA\}$ , where NA denotes there is no relation between a pair.

In order to represent the candidate relation pair in an input sequence, we replace the relevant entity names with their semantic types followed by standard preprocessing step for RE. “We further show that PROTEIN\$ directly interacts with PROTEIN\$ and Rpn4.” is an example of input representation to the RE model.

We fine-tune SciBERT model Beltagy et al. (2019) followed by a linear classification layer added on top to predict the relation type of a candidate pair. In other words, given sequence  $s_i = (w_1, \dots, w_n)$ , where  $w_i$  is the  $i$ -th token in the sequence, we feed  $s_i$  into the SciBERT, and retrieve the hidden state representation of the sequence (CLS) along with the words.

$$(h_{\text{CLS}}, h_1, \dots, h_n) = \text{SciBERT}(w_1, \dots, w_n)$$

Where  $h_i$  is the representation of  $i$ -th token into a  $d$  dimensional space. As typical with BERT Devlin et al. (2019), we use  $h_{\text{CLS}}$ , corresponds to the aggregate representation of a sequence, as input to classification layer. Then, a softmax layer is added to output labels for the sentence.

$$z_i = \text{softmax}(Wh_{\text{CLS}})$$

Where  $W$  is the learnable parameters for linear layer, and  $z_i$  is the probability vector assigned to each relation types.

### 4.3 Methodology

The intuition behind our approach, MERIT, stems from the fact that due to power-law distribution, a data point  $x_i$  may have similar features (based on distance-based metrics) to  $k$  other data points  $X_k = \{x_j\}$  in the corpus, so-called local community. In contrast, there might be  $m$  unsimilar data points due to long-tail distribution, where  $m \gg k$ . The question that we want to answer is how to maximize the annotation effort provided by an expert who has a limited labeling budget?

To this end, we hypothesize that if such a local community exists for a data point  $x_i$ , we can augment the expert-annotated data by transferring the label of the  $x_i$  to its whole community  $X_k$ . As a result, we augment the training labels by utilizing expert supervision to generate high-quality weak labels, which can be further used in boosting the performance of any supervised RE models.

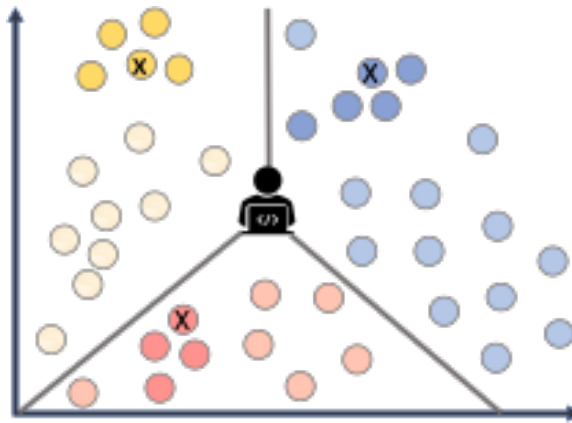


Figure 4.1. Label augmentation through local community search. The data points annotated by X denote strong label, while data points highlighted with bold color are the weak labels.

There are two key parameters in our approach that provide high-quality weak labels: 1) feature representation, and 2) distance threshold. Figure 4.1 illustrates the proposed approach. It depicts the local community in 2-dimensional space.

There are several choices to encode features by semantic/static-based embeddings Devlin et al. (2019); Church (2017) to provide representation for a sequence. However, these representations are too general and less concerned with the target labels. Ideally, we aim to generate an embedding that captures a semantic representation which is more correlated to the target labels. To this end, we apply a dependency parse tree on a sequence to extract the shortest dependency path (sdp) between two entities. As it has been shown in previous work Li et al. (2019); Zhang et al. (2018), sdp can boost the performance of RE models and provide strong hints about the relation between entities.

Let  $T$  be a rooted parse tree corresponding to sequence  $s_i$ . Given a pair of entities  $(e_{\text{subj}}, e_{\text{obj}})$ , sdp can be defined as a minimum set of tokens that can be reached from  $e_{\text{subj}}$  to  $e_{\text{obj}}$  through the dependency tree  $T$ . For example, in the sentence “Chemical caused a dose-dependent reduction in plasma Gene.”, the terms (caused, reduction) are the sdp tokens between entities (Chemical, Gene).

In order to integrate the sdp information into the feature space, we take the average embedding of sdp tokens and concatenate them with entity representations based on SciBERT Beltagy et al. (2019).

$$h_i = \text{concat} \left( \frac{1}{k} \sum_{t \in \text{sdp}} h_t, h_{\text{subj}}, h_{\text{obj}} \right)$$

Where  $h_i$  corresponds to the final representation for sentence  $s_i$ .

The second key decision in our framework is to compute local communities for data point  $x_i$  according to the above feature representation. To this end, we consider a threshold  $\theta$ , under which the  $x_j$  cannot be a local community of  $x_i$ .

For optimized calculations of local community around each data point, we cluster the data using the Kmeans clustering algorithm. Since we just search for local communities around each data point, Kmeans clustering reduces the search space by an order of magnitude. Finally, we compute the cosine similarity as the distance metric to identify the local communities. The equation below demonstrates the corresponding function:

$$\text{LocCommunity}(x_i, x_j) = \begin{cases} x_j & \text{if distance}(x_i, x_j) \leq \theta \\ \emptyset & \text{otherwise} \end{cases}$$

## 4.4 Experiments

First, we describe the characteristics of the datasets that we used. Next, we explain the experimental design. Finally, we discuss the results.

### 4.4.1 Datasets

We evaluate our approach on three benchmark biomedical RE datasets. The characteristics of these datasets have been shown in Table 4.1. Followed by previous works Krallinger et al. (2017); Herrero-Zazo et al. (2013), for ChemProt and DDI datasets, we use the same train, development, and test splits during model development. In addition, for the PPI dataset, we utilize AIMed corpus Bunescu et al. (2005) and performed 5-fold cross-validation due to the lack of standard train and test split. ChemProt and DDI tasks are multi-class classification problems. For ChemProt, there are 6 different relation types to capture the interaction between chemical and protein. In addition, the DDI contains 5 relations that correspond to the interaction between drugs. The labels for the DDI task are advice, effect, int, mechanism, and negative. PPI task is a binary classification problem to extract human protein interactions.

Table 4.1. Statistics of the dataset used for label augmentation.

Dataset	Train	Validation	Test	#relations
ChemProt	18k	11.2k	15.7k	6
DDI	22k	5.5k	5.7k	5
PPI	5.2k	-	583	2

#### 4.4.2 Evaluation Metric and Experiment Design

We report standard precision (P), recall (R), and F1-score (F1) for binary classification, and micro-P, micro-R, and micro-F1 for multi-class classification. The evaluation metrics are as follows:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F1 = \frac{2 \cdot P \cdot R}{P + R}$$

For multi-class classification, the TP in micro-P and micro-R are the number of all positive class (we consider NA type as the negative label). We fine-tuned SciBERT for RE task. We set maximum the sequence length to 200, batch size to 16, distance similarity threshold to 0.9, learning rate to 2e-5, and epochs to 10. The remaining hyperparameters were used as their default values.

#### 4.4.3 Results

We compared the effectiveness of our approach against the random sampling baseline (RS). To this end, we run experiments for different labeling budget sizes [100, 200, 500] to acquire expert annotations. In both experiments, we randomly sample from the corpus and train the RE model on the collected samples. Compared to the RS baseline, MEIRT has the added benefit of leveraging the weak labels along with the strong labels. This experiment shows the effectiveness of MERIT as an extension to any supervised RE model. Figure 4.2 demonstrates the significant improvement of our approach over RS baseline. Due to the random selection, we conduct all the experiments in 3 independent runs and report the

average performance. The results show that in a scenario that limited annotated data is available (100), there is no learning for RS baseline model. However, in this setting, our approach can boost the final performance by  $\sim 0.20$  F1 scores compare to RS baseline. In addition, as we increase the hand-labeled data (up to 500), we still outperform the baseline by up to 36% increase in the final F1 score. This shows that the weak labels do not damage the performance in larger budget sizes, but further are very informative to the target task.

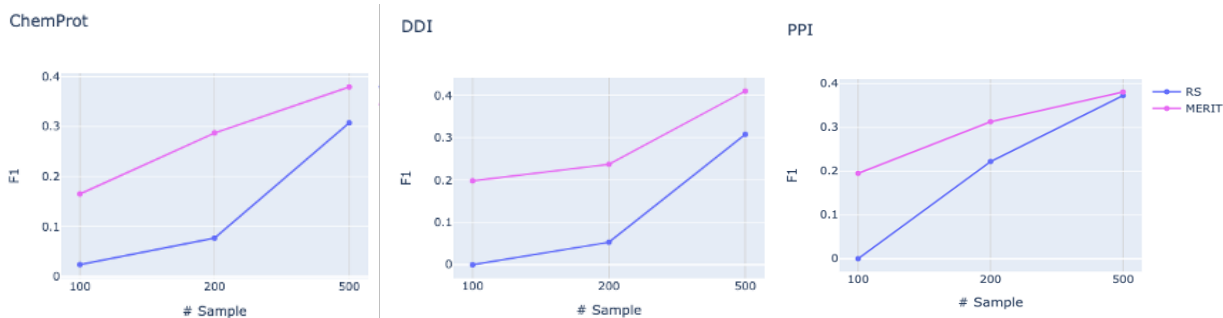


Figure 4.2. Comparison of our approach with RS baseline on three benchmarks biomedical RE datasets.

#### 4.4.4 Ablation Study

To further evaluate the effectiveness of each parameter in our approach, we perform an ablation study on the impact of distance threshold and feature representations. We conduct experiment on 200 labeling budget with different threshold values in the range of  $\{0.7, 0.75, 0.80, 0.85, 0.9, 0.95\}$  on three datasets. As Figure 4.3 illustrates, as the threshold decreases to 0.7, we allow more weak labels. This in turn leads to more noise in the labels, hence damaging the performance. We found that 0.90 is the optimal threshold value for all three datasets to keep the balance between weak and strong labels, thereby increasing the performance by up to 0.2 F1 scores. This parameter can be further optimized during the learning process for future research directions.

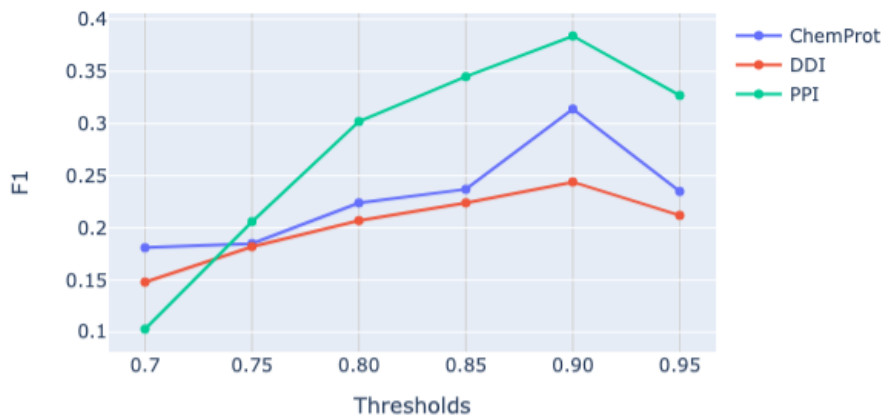


Figure 4.3. The impact of threshold on the final performance. We use 200 labeling budgets to perform this experiment.

Moreover, we explored the impact of several feature representations in order to calculate the distance function. We performed this experiment on ChemProt dataset with a labeling budget of 200. We consider the following different choices:

- CLS ( $L$ ), where  $L$  is the final embedding dimension, is an aggregate representation of the tokens in a sentence
- ent-avg ( $L$ ), takes the average embedding of the entities in a sentence
- ent-sdp-avg ( $L$ ), takes the average embedding of the entities and sdp tokens
- ent-concat ( $2L$ ), concatenates the embedding of entities in a sentence
- ent-words-between ( $3L$ ), concatenates the embedding of entities along with the average representation of all the words between two entities
- ent-concat-sdp-avg ( $3L$ ), concatenates the embedding of entities along with the average representation of sdp tokens

As it is shown in Figure 4.4, both ent-concat-sdp-avg and ent-sdp-avg representations outperform non-sdp-based representations. This highlights the fact that sdp provides necessary information for the distance function to combine similar examples in computing the

local community. The ent-concat-sdp-avg representation increases the final F1 score by 163% over CLS and 52% over ent-avg and ent-concat embeddings. In addition, the result for entity-words-in-between-based embedding shows that adding unnecessary context is not beneficial for training. However, it can still outperform the baseline CLS by 27% increase in F1 score. In addition, another finding is that averaging the embedding usually performs slightly less than concatenation (as illustrated in Figure 4.4).

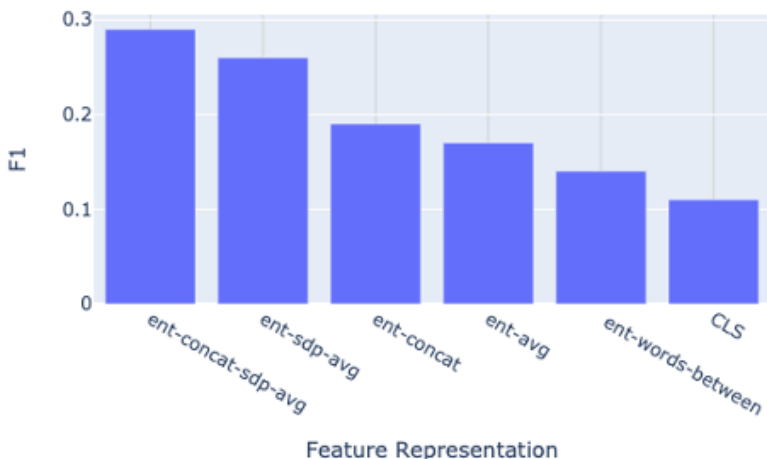


Figure 4.4. Comparison with different feature representations on ChemProt dataset with 200 labeling budgets.

## 4.5 Conclusion

We proposed MERIT, a simple yet effective approach to address gold-labeled data deficiency in Biomedical RE models. We maximized the expert supervision by using a heuristic to search for local communities around strong samples. MERIT utilizes Shortest Dependency Path in between entities as a representation that closely captures the target relation. Therefore, it is likely that sentences with similar representations have the same relation label. We used cosine similarity as a measure of distance function and tune a threshold to create local communities around each strong label. We improved the performance of the

RE models by 2x in F1, when trained on both weak and strong labels. We demonstrated the impact of MERIT on three benchmark biomedical RE datasets.

## CHAPTER 5

### LEVERAGING SHORTEST DEPENDENCY PATHS IN LOW-RESOURCE BIOMEDICAL RELATION EXTRACTION

#### 5.1 Introduction

Biomedical Relation Extraction (RE) plays a pivotal role in structuring unstructured medical texts, enabling the construction of knowledge graphs Bairoch & Apweiler (1997); Wishart et al. (2006) and the extraction of complex relationships between biomedical entities such as drugs, proteins, and genes Köhler et al. (2000); Von Mering et al. (2002); Wilkinson (2005). Effective RE aids in the discovery of new drug interactions and biological pathways, critical for advancing medical research and clinical decision-making.

Despite advancements through supervised RE methods Wei et al. (2019); Lee et al. (2020); Zhou et al. (2021), their efficacy is often limited by the scarcity of labeled biomedical data. Several approaches, such as weak supervision Mintz et al. (2009) and semi-supervised learning (SSL) Ouali et al. (2020), have been developed to address the challenges of limited training data through leveraging unlabeled data. More recently, in-context learning techniques using Large Language Models (LLMs) have emerged, requiring significantly less labeled data Liu et al. (2021); Rubin et al. (2021).

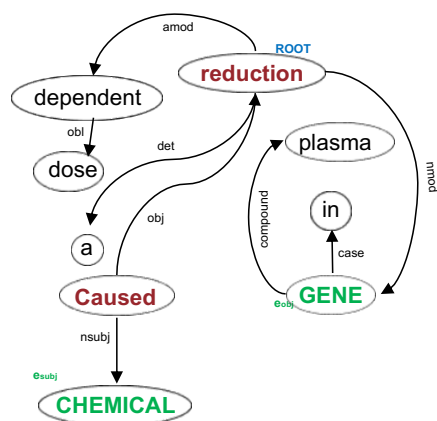
Weak supervision, for instance, utilizes heuristics, rules, or distant supervision to generate noisy labels for unlabeled data, a technique pioneered by Mintz et al. (2009). Although this method enhances the volume of trainable data, it also introduces label noise. Other strategies, such as those developed by Qu et al. (2018) and Zhou et al. (2020), employ linguistic patterns like SDP (Shortest Dependency Path) tokens or frequent phrases to formulate labeling rules, reducing the need for manual annotation yet increasing the hidden costs of rule derivation.

SSL utilizes both a limited pool of labeled data and a larger volume of unlabeled data to enhance learning models Ouali et al. (2020). Common SSL strategies such as self-training rely heavily on predictors that impute labels on unlabeled examples, which are then used to retrain the models Rosenberg et al. (2005). Variants of self-training include dual training, where a secondary predictor retrieves relevant unlabeled instances Lin et al. (2019), and Gradient Imitation Reinforcement Learning (GIRL), which optimizes the correlation between gradients of labeled and unlabeled data to enhance performance Hu et al. (2021).

The challenge in self-training is limited labeled data leads to low-quality label imputations, as the reliance on the predictor impedes learning a powerful model. Likewise, graph-based SSL methods, such as label propagation, utilize lexical and syntactic features to propagate labels among closely situated data points Zhou et al. (2003); Chen et al. (2006). However, determining the correct distance metric and neighborhood thresholds necessary for identifying relevant relational patterns within data remains an open research question. This issue also affects in-context learning, where selecting relevant examples from a constrained dataset for task demonstrations also requires precise distance metrics.

This study aims to address two pivotal questions: What is an effective representation for defining a suitable distance metric in low-resource settings, and how can we reduce the model’s dependence on imputed labels to boost RE accuracy in SSL scenario? To tackle the first question, we propose employing the SDP between entities to process the encoder output and compute the SDP representation for RE. SDP, derived from dependency parse trees, identifies the minimal syntactic dependencies essential for connecting entities, providing valuable hints about their relationships Li et al. (2019); Zhang et al. (2018); Liu et al. (2013) (see Figure 5.1).

For the second question, we advocate using a nearest neighbor approach instead of relying solely on model-based imputations. This method carefully propagates labels to nearby data points based on a specifically defined SDP-based distance metric, thereby allowing



*Chemical caused a dose-dependent reduction in plasma Gene*

Figure 5.1. A dependency parse tree on a biomedical sentence and its shortest dependency path (SDP) tokens (shown in red) between subject (CHEMICAL) and object (GENE) entities.

for the integration of soft labeling techniques that account for the uncertainty and noise in label imputation.

By addressing these research questions and conducting extensive experiments across three biomedical RE benchmarks, we aim to develop a versatile strategy compatible with any standard RE architecture and SSL algorithm. This approach is designed to deliver accurate results in various low-resource environments, encompassing supervised, SSL, and in-context learning scenarios. In summary, the contributions of this paper are:

- We propose utilizing SDP to calculate SDP representation of entity pairs for RE, improving accuracy in various low-resource settings such as supervised learning and in-context learning.
- We use SDP representation to calculate SDP-based distance metric between RE examples. We use this distance metric to support two types of SSL algorithms. We experimentally evaluate on biomedical text the usefulness on the distance metric.
- Our extensive experiments on three key biomedical RE benchmarks confirm the efficiency of our proposed method in low-resource settings.

## 5.2 Background and Task Formulation

In this section, we introduce the relevant concepts and formally define the RE task.

**Shortest Dependency Path (SDP).** Let  $\mathcal{T}$  be a dependency parse tree corresponding to sequence  $s$  representing the syntactical relationship between words in a sentence. In a dependency parse tree, words are represented as nodes, and the relationships between words are represented as directed edges. Given a pair of entities  $(e_s, e_o)$ , SDP is defined as the minimum set of tokens that can be reached from  $e_s$  to  $e_o$  through the dependency tree  $\mathcal{T}$ . Figure 5.1 shows an example of a parse tree, where the extracted SDP tokens between a pair of entities (Chemical and Gene) are highlighted in red. We need to highlight that we don't consider the relation dependency between words in this study.

**Relation Extraction (RE).** Given sentence  $s = (w_1, w_2, \dots, w_n)$ , a subject entity  $e_s$ , and an object entity  $e_o$ , the RE task is to predict the relation label  $r \in \mathcal{R}$  of triple  $x = (s, e_s, e_o)$ , where  $\mathcal{R}$  is a union of a predefined set of relation types and None, referring to no relation or other type of relation.

**Semi-Supervised RE.** This approach utilizes a small set of labeled examples  $\mathcal{D}_L = \{(x_i, r_i)\}_{i=1}^{N_l}$  and a larger set of unlabeled examples  $\mathcal{D}_U = \{(x_j)\}_{j=1}^{N_u}$  to train a classifier model  $f_\theta$ . The model aims to fit the labeled data while also leveraging the unlabeled data to improve overall accuracy.

In the following subsections, we refer to  $x = (s, e_s, e_o)$  as an RE example and  $r$  as the RE label. In addition, we assume the entity mentions can be identified using external tools and the set of relation types  $\mathcal{R}$  is defined.

## 5.3 Methodology

### 5.3.1 SDP Representation

In this section, we propose the SDP representation for RE. RE is defined as a text classification problem. As an encoder, we utilize the BERT neural network architecture

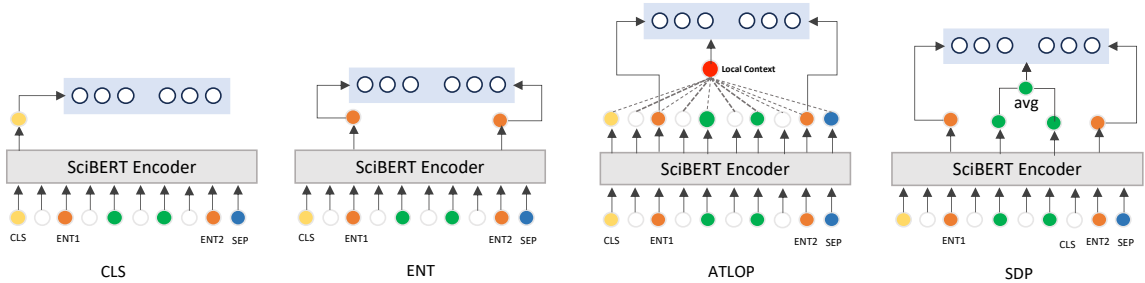


Figure 5.2. Comparison between different representations to fine-tune a RE using a linear layer on top of an encoder. Orange corresponds to the entity representation. Green corresponds to the SDP tokens between entities.

Devlin et al. (2019). BERT takes a sentence of tokens as input and produces embedding vectors for the entire sentence and each individual token, denoted as  $H : [h_{cls}, h_1, \dots, h_n] = \text{BERT}(w_1, \dots, w_n)$ , where  $w_i$  represents the  $i$ -th token in the sentence,  $h_i$  is its embedding, and  $h_{cls}$  is the sentence embedding.

This sequence of vector embeddings can be used in multiple ways as an input to the classification head. As illustrated in Figure 5.2, we first present several baseline approaches for using a sequence of embedding vectors. Then, we propose how to calculate the SDP embedding.

**CLS:** BERT CLS embedding is used as an input to the RE decoder. This representation has been commonly used for many downstream text classification tasks, including RE Lee et al. (2020); Wei et al. (2019). We denote the dimension of this token as  $L$ , which corresponds to the length of  $h_{cls}$ .

$$\text{CLS}_{\text{rep}} = h_{cls} \quad (5.1)$$

**ENT:** Followed by the best setup in Baldini Soares et al., 2019, we concatenate embeddings of the two entity tokens. If an entity consists of multiple tokens we only consider the first token.

$$\text{ENT}_{\text{rep}} = h_s \oplus h_o \quad (5.2)$$

where  $\oplus$  denotes the concatenation. The dimension of this representation as  $2L$  which corresponds to  $h_s$  and  $h_o$ .

**ATLOP:** Based on Zhou et al. (2021), we combine the entity embeddings using a vector called the local context. This vector contains pertinent information related to both entities. Let’s consider  $A_s$  and  $A_o$  as the self-attention matrices for entities  $e_s$  and  $e_o$  from the last layer in BERT, where  $A \in R^{H \times l \times l}$ .  $A_{ijk}$  represents attention from token  $j$  to token  $k$  in the  $i_{th}$  attention head, while  $A_i^E \in R^{H \times l}$  denotes attention from the  $i_{th}$  entity to all tokens. We locate the local context that is important to both  $e_s$  and  $e_o$  by multiplying their entity-level attentions, and obtain the localized context embedding  $c(s, o)$  by:

$$\begin{aligned}
 A^{(s,o)} &= A_s^E \odot A_o^E, \\
 q^{(s,o)} &= \sum_{i=1}^H A_i^{(s,o)}, \\
 a^{(s,o)} &= q^{(s,o)} / 1^\top q^{(s,o)}, \\
 c^{(s,o)} &= H^\top a^{(s,o)}
 \end{aligned} \tag{5.3}$$

Where  $H$  is the contextual embedding derived from BERT. To construct the final embedding for an instance, we concatenate the local context embedding with the embeddings of the entity pair:

$$\text{ATLOP}_{\text{rep}} = h_s \oplus c^{(s,o)} \oplus h_o \tag{5.4}$$

The resulting representation has length  $3 \times L$ .

**SDP<sub>rep</sub>:** Our main contribution is the use of SDP to enrich the embeddings by focusing on syntactic paths that are most indicative of relations. Biomedical sentences use complex terminology can be quite long, but they follow a relatively structured grammar. We hypothesize that the SDP between pairs of entities in biomedical documents is very likely to contain key information in revealing the relation type. To this end, we employ SDP to direct our attention mechanism within BERT encoder. By averaging the embeddings of the tokens that constitute the SDP and concatenating these with the embeddings of the starting

and ending tokens of the entities involved, we form a meaningful representation which can be formulated as follow:

$$SDP_{rep} = h_s \oplus \left( \frac{1}{|SDP|} \sum_{w_i \in SDP} h_i \right) \oplus h_o, \quad (5.5)$$

where  $|SDP|$  denotes the number of SDP tokens. The dimension of SDP representation is  $3 \times L$ , which blends syntactic precision with semantic richness.

In the RE model, the decoder is a single classification layer  $W \in R^{K \times L^*}$  where  $L^*$  is the dimension of the input representation ( $L^* = 3L$  for SDP representation) and  $K$  is the number of relation types. Importantly, this decoder can be replaced with any off-the-shelf RE architecture and is placed on top of the processed encoder representation. In addition, the SDP representation can be integrated separately to define a distance metric such as in in-context learning or retrieval scenarios, enhancing the model’s applicability to a broader range of tasks that require a nuanced understanding of entity relationships. The classification of the RE task given a representation is performed by:

$$p(r|x) = \text{softmax}(V_x W^T) \quad (5.6)$$

where  $V_x$  is the output vector from the last layer of the encoder corresponding to the SDP-enhanced representation for input  $x$ , and  $W$  represents the weight matrix of the classification layer. The loss for training is computed as the cross-entropy between  $p(r|x)$  and the true relation labels.

### 5.3.2 SDP in Semi-Supervised RE

This section explores the integration of the Shortest Dependency Path (SDP) with nearest neighbor-based label propagation in an extreme low-resource SSL setting. Our approach is designed to effectively utilize a minimal set of labeled data points, thereby eliminating the need for a large labeled for training and validation set.

Our primary goal is to leverage the SDP representation in conjunction with nearest neighbor techniques to compute distances between RE examples. This enables precise label propagation and reduces reliance on predictors for generating pseudo-labeled data.

We developed an SSL approach that combines graph-based SSL and self-training. Similar to graph-based SSL, our algorithm propagates labels to closely neighboring unlabeled data. Unlike traditional graph-based methods, it meticulously restricts label propagation to only the nearest neighbors rather than all unlabeled data (Zhu & Ghahramani (2002)). This constraint is essential in low-resource settings, as it minimizes the noise introduced by including broader sets of unlabeled data and preserves the influence of the scarce labeled data available in the training process. From the self-training perspective, while our approach utilizes the predictor to extract representations, it does not rely on it to generate pseudo-labels. This is because it is challenging to develop a reliable and unbiased predictor with limited labeled data. Instead, we utilize these representations to compute soft labels, which are determined based on the proximity of unlabeled RE examples to their labeled counterparts. This technique indirectly employs the predictor, enhancing the label quality without the direct use of potentially biased pseudo-labels. Prior research shows that soft labels result in a more robust classifier (Sajjadi et al. (2016)) by informing the training process about the quality of imputed labels. To compute soft labels for unlabeled examples, we follow the steps below:

- Find the nearest neighbor of the unlabeled example among the labeled examples.
- Calculate the cosine similarity( $d$ ) between the unlabeled example and its top-k nearest neighbors (labeled data).
- Aggregate the cosine similarities for each class type
- Compute the soft labels for each class type by normalizing the aggregated similarities using softmax.

To optimize the RE model on soft labels, we use Noise Aware Cross Entropy loss (Equation 5.7), which is computed for each class separately, and then the losses are summed together,

$$\mathcal{L} = \sum_{i=1}^N \sum_{c=1}^K y_{i,c} \log \hat{y}_{i,c}, \quad (5.7)$$

where  $N$  is the total number of examples, and  $y_{i,c}$  and  $\hat{y}_{i,c}$  represent the ground truth and predicted soft label for example  $i$  and class  $c$ . The algorithm is executed iteratively, with each cycle incorporating a limited number of additional soft labels. We specify a fractional amount of unlabeled data as validation set and monitor the predictor’s fluctuations to determine convergence. Convergence is achieved when the prediction variation on a validation set is less than 5% between iterations, or when the maximum number of iterations is reached. It is important to note that our algorithm does not rely on ground truth validation data, using instead the stability of predictions as a stopping criterion.

This semi-supervised approach, centered on the strategic use of SDP and soft labeling along with nearest neighbor-based propagation, is designed to enhance the efficiency of RE models in SSL settings constrained by extreme limited labeled data.

## 5.4 Experiments

### 5.4.1 Dataset

We evaluate the effectiveness of our method on three public biomedical relation extraction datasets retrieved from PubMed database. The statistics of these datasets are shown in Table 5.1. 1) **ChemProt** Krallinger et al. (2017) consists of 1,820 PubMed abstracts with chemical-protein interactions annotated by domain experts and was used in the BioCreative VI text mining chemical-protein interactions shared task. 2) **DDI** Segura-Bedmar et al. (2013) contains MedLine abstracts on drug-drug interactions as well as documents describing drug-drug interactions from the DrugBank database. 3) **PPI** Bunescu et al. (2005)

utilizes AIMed corpus to automatically extract interaction relations of protein-protein pairs affected by genetic mutations.

Table 5.1. Statistics of each dataset

<b>Dataset</b>	<b>Train</b>	<b>Validation</b>	<b>Test</b>	<b>#Relations</b>
ChemProt	18k	11.2k	15.7k	6
DDI	22k	5.5k	5.7k	5
PPI	5.2k	521	583	2

#### 5.4.2 Compared Methods

To perform experiments, we first compared our SDP-based finetuning strategy with several supervised RE baselines. Then, we adopted the best-performing baseline as the RE classifier model to explore the impact of SDP in SSL baselines and in-context-learning. In all the experiments, we applied SciBERT Beltagy et al. (2019) as the encoder. We performed all the experiments under a very limited budget for labeled data and abundant unlabeled data. We denote SUP-RE<sub>sdp</sub> and SSL-RE<sub>sdp</sub> as supervised and semi-supervised variants of SDP in the remaining subsections.

**Supervised Baseline Methods:** The goal is to compare the performance of different RE architectures (as discussed in Section 5.3.1) in a supervised setting and show that fine-tuning using SUP-RE<sub>sdp</sub> achieves a better performance compared to the existing approaches. These approaches are CLS, ENT, ATLOP. This experiment utilizes limited labeled data to explore the best-performing RE model (in our case is 500).

**Semi-Supervised Baseline Methods:** The goal of this experiment is to compare the superior performance of SSL-RE<sub>sdp</sub> with predictor-based SSL. To ensure fair comparison with SSL-RE<sub>sdp</sub>, we used the best-supervised baseline from Table 5.2, which was SUP-RE<sub>sdp</sub>, as RE model. We applied the following SSL methods on  $\mathcal{D}_L \cup \mathcal{D}_U$ :

(1) Label Propagation Zhu & Ghahramani (2002), which is a graph-based algorithm that iteratively updates the label probability in  $\mathcal{D}_U$  by matrix multiplication ( $TR$ , where  $T$  is a  $n \times n$  weighted adjacency matrix (pairwise relations between labeled and unlabeled data) and  $R$  is  $n \times C$  class probability matrix). (2) Self-Training Rosenberg et al. (2005), which iteratively expands  $\mathcal{D}_L$  by using the most confident (above  $\tau$ ) predictor’s prediction among  $\mathcal{D}_U$ . (3) DualRE Lin et al. (2019), which is a dual training algorithm that utilizes a learning-to-rank model as a dual module to retrieve the relevant instances from  $\mathcal{D}_U$  for a given relation. (4) RE-Ensemble, which replaces the dual module in DualRE Lin et al. (2019) with the same predictor in the primal module, with a different random initialization. RE-Ensemble imputes the labels based on the agreement of the two modules.

We also provide SUP-RE<sub>sdp</sub> as a supervised baseline, which can also serve as a few-shot baseline since it is only trained on limited labeled data without access to unlabeled data. In addition, we report SSL-RE<sub>sdp</sub>-1-Iter, which is the same as SSL-RE<sub>sdp</sub>, but only uses one iteration to perform imputation.

### 5.4.3 Experimental Setting

**Implementation:** We implemented all the baselines using Pytorch. For DualRE and RE-Ensemble Lin et al. (2019), we replaced the Position-aware Recurrent Neural Network that was used originally in Zhang et al. (2017) with SUP-RE<sub>sdp</sub>. The source code for these baselines can be found here<sup>1</sup>. In addition, we used the code provided by Pedregosa et al. (2011) to apply label propagation algorithm.

**Training details:** We adopted SciBERT as the encoder for all the experiments and update all the parameters. For supervised finetuning, we add one linear layer followed by softmax to perform classification. We use the following set of hyper-parameters as suggested in Devlin et al. (2019):

---

<sup>1</sup> DualRE. <https://github.com/INK-USC/DualRE>. Accessed 11 July 2023.

- Transformer Architecture: 12 Layers, 768 hidden dimension, 6 heads
- Learning rate:  $3e-5$
- Weight initialization: SciBERT-base
- Batch size: 32
- Optimizer: adam
- Training epochs: 5
- Maximum sequence length: 200

We used 1 GPU, Tesla V100-SXM2, for training. We applied SciSpacy Neumann et al. (2019) dependency parser to our corpus to retrieve the SDP tokens for an entity pair.

For SSL experiments, we kept the same hyper-parameters. We impute labels to the top-5 unlabeled data. A similar strategy is applied to retrieve labeled examples for each unlabeled data to compute soft-labels.

In Self-Training, since we use the RE model to provide predictions on unlabeled data, we set the threshold for the most likely class to be above 0.90. However, since the majority of the predictions were overconfident based on the validation results, resulting in imputing noisy labels, we only select the top 100 in the augmented set. In Label Propagation implementation based on Pedregosa et al. (2011), we chose KNN as kernel function, and set the  $K$  to 5, which specifies the number of closest labeled instances to include in the label propagation process for each unlabeled instance. For DualRE and RE-Ensemble, we followed the default hyperparameters mentioned in Lin et al. (2019). We only leveraged 50% of  $\mathcal{D}_U$ , and used the default confidence thresholds  $\alpha = 0.5$  and  $\beta = 2$  predictor and retrieval modules, respectively. We applied the same convergence criteria as in SSL-RE<sub>sdp</sub> for self-training and dual training.

**Evaluation metric:** Following the previous work in RE Lin et al. (2019); Zhou et al. (2020), we report micro-F1 as the most important evaluation metric. It provides an evalua-

tion of the model’s ability to simultaneously capture precision and recall across all classes. We ignored correct predictions of None in micro score calculation.

## 5.5 Results

We conducted each experiment over three different independent sets of labeled data and reported the mean performance.

### 5.5.1 Comparison With Supervised Baselines

Table 5.2 demonstrates the performance of different supervised RE architectures under 500 training budget. SUP-RE<sub>sdp</sub> approach achieves higher accuracy compared to the CLS, ENT, and ATLOP architectures. This can be attributed to the explicit guidance provided by SDP, which directs the predictor to focus on tokens relevant to the target label in biomedical settings where limited labeled data exists.

Among the approaches considered, the CLS representation exhibits the lowest performance. This could be due to the fact that it is sentence-level representation, having less relevant information for entities.

When compared to ATLOP, SUP-RE<sub>sdp</sub> appears to have slightly better F1 score across all datasets. This indicates that the local context pooling mechanism in ATLOP does not capture dependencies as accurately as SUP-RE<sub>sdp</sub>. Furthermore, SUP-RE<sub>sdp</sub> slightly outperforms ENT-based fine-tuning on the DDI and ChemProt datasets, while delivering comparable performance on PPI.

To statistically validate the performance differences observed, a Repeated Measures ANOVA was conducted for each dataset. This analysis confirmed the significance of the observed variations in performance, with the p-value for DDI at  $p = 0.0030$ , for ChemProt at  $p = 0.0028$ , and for PPI at  $p = 0.0025$ . The consistency of these statistically signifi-

Table 5.2. Performance of different RE finetuning architectures when trained using 500 labeled data. The average F1 performance is reported over 3 independent runs.

	DDI			ChemProt			PPI		
	P	R	F1	P	R	F1	P	R	F1
CLS	0.24	<b>0.77</b>	0.36	0.16	0.52	0.24	0.37	0.82	0.51
ENT	0.28	0.74	0.40	<b>0.23</b>	0.62	0.33	0.45	<b>0.84</b>	0.59
ATLOP	0.28	0.74	0.41	<b>0.23</b>	0.60	0.33	0.45	0.82	0.58
<b>SUP-RE<sub>sdp</sub></b>	<b>0.29</b>	<b>0.77</b>	<b>0.42</b>	<b>0.23</b>	<b>0.63</b>	<b>0.34</b>	<b>0.46</b>	0.80	<b>0.59</b>

cant results supports the superior efficacy of the SUP-RE<sub>sdp</sub> approach across all examined datasets, reaffirming its selection for further analysis.

Considering the slightly better performance of SUP-RE<sub>sdp</sub>, as shown in Table 5.2, and the statistical confirmation of its superiority through ANOVA testing, we have selected it as the RE model for the subsequent subsections. These findings emphasize the importance of methodological selection and highlight the benefit of leveraging SDP-guided approaches in low-resource settings for RE tasks.

**5.5.1.1 Comparison With Non-Encoder Baselines** This experiment evaluates our supervised relation extraction (RE) method, which integrates Shortest Dependency Paths (SDP) and BERT-based representations, against traditional non-encoder baselines utilizing SDP or dependency trees as graph kernels for relation extraction. The fundamental principle of these kernel methods is to assess the similarity between two sentences by examining how closely their structural patterns align. These kernels operate in conjunction with Support Vector Machines (SVM) to classify sentences. Our analysis focuses on the Protein-Protein Interaction (PPI) dataset due to the availability of extensive kernel method benchmarks. We adopted the experimental setup from Tikk et al. (2010) to ensure a consistent comparison with the kernel methods listed in Table 2 of their study. All experiments were conducted using 10-fold cross-validation on the full PPI dataset, corresponding to the AIMed results in Table 2 of Tikk et al. (2010). Following their recommendations, we

implemented entity blinding to prevent the influence of named entity recognition problems and to highlight entity locations to the classifier. Our results are compared with a range of kernel methods as detailed in Table 5.3:

**Edit Distance Kernel (edit)** Erkan et al. (2007): This kernel calculates the similarity by measuring the edit distance between the shortest paths connecting protein names within a dependency tree. The similarity is determined by the minimum number of edit operations—deletions, insertions, or substitutions required to make one path identical to the other, normalized by the length of the longer path.

**Cosine Similarity Kernel (cosine)** Erkan et al. (2007): This method computes the cosine similarity between vectors representing the shortest paths in a dependency parse tree between pairs of entities. It quantifies the number of common terms along these paths, adjusted for path length.

**All-Paths Graph Kernel (APG)** Airola et al. (2008): APG considers all possible path lengths within the dependency parse and surface word sequence, assigning greater weight to paths closer to the shortest path between entities, thereby reflecting dependency proximity.

**k-Band Shortest Path Spectrum Kernel (kBSPPS)** Palaga (2009): This kernel extends the analysis beyond the shortest dependency path to include nodes within a specified k-band distance, enriching the contextual data for relationship extraction.

**Other Kernels:** We further compare our method against kernels that utilize syntax tree representations of sentences, such as the Subtree kernel (ST) Smola & Vishwanathan (2002), Subset tree kernel (SST) Collins & Duffy (2001), Partial tree kernel (PT) Moschitti (2006), and Spectrum tree kernel (SpT) Kuboyama et al. (2007).

Table 5.3 showcases a comparative analysis between various non-encoder-based kernel methods and our SDP-based approach for relation extraction. Notably, our method, SUP-RE<sub>sdp</sub>, significantly outperforms the other models in precision (P), recall (R), and F1 score, achieving 81.21% precision, 78.0% recall, and an F1 score of 79.4%. This demon-

Table 5.3. Comparative analysis of non-encoder based kernel methods using Shortest Dependency Paths (SDP) against our supervised method, which also utilizes SDP for representation. Performance metrics are evaluated using a 10-fold cross-validation on the PPI dataset.

	<b>P</b>	<b>R</b>	<b>F1</b>
<b>Syntax Tree Kernel</b>			
ST Smola & Vishwanathan (2002)	40.3	25.5	30.9
SST Collins & Duffy (2001)	42.6	19.4	26.2
PT Moschitti (2006)	39.2	31.9	34.6
SpT Kuboyama et al. (2007)	33.0	25.5	27.3
<b>SDP kernel</b>			
edit Erkan et al. (2007)	68.8	27.7	39.0
cosine Erkan et al. (2007)	43.6	39.4	40.9
APG Airola et al. (2008)	62.9	48.9	54.7
kBSPS Palaga (2009)	50.1	41.4	44.6
<b>SUP-RE<sub>sdp</sub> (Ours)</b>	<b>81.2</b>	<b>78.0</b>	<b>79.4</b>

strates a marked improvement over traditional non-encoder methods like the APG kernel, which has the next highest F1 score of 54.7% but with a substantially lower recall. The kBSPS, while competitive to APG, still trails our method with an F1 score of 44.6%. The substantial lead in performance metrics highlights the effectiveness of integrating SDPs with BERT-based representations, providing evidence that our LLM-based representation using SPD captures complex semantic relationships more effectively than conventional kernel methods.

### 5.5.2 Comparison With Semi-Supervised Baselines

Table 5.4 shows the result of our approach compared to SSL baselines and few-shot supervised baseline (SUP-RE<sub>sdp</sub>). According to the results, one can observe that SSL-RE<sub>sdp</sub> outperforms all of the baselines across all datasets, which demonstrates the effectiveness of our framework versus SSL baselines.

Table 5.4. The F1 comparison of  $\text{SSL-RE}_{sdp}$  versus SSL baselines.  $\text{SUP-RE}_{sdp}$  serves as the supervised lower bound. The lower/upper bound for F1 metrics is 0/1. We report the average performance across three independent runs.

	50			100			200			500		
<b>DDI</b>	P	R	F1	P	R	F1	P	R	F1	P	R	F1
$\text{SUP-RE}_{sdp}$	0.15	0.50	0.23	0.21	0.58	0.31	0.24	0.65	0.35	0.32	0.82	0.46
LabelPropagation	0.076	0.42	0.15	0.13	0.59	0.21	0.18	0.76	0.25	0.24	<b>0.88</b>	0.35
DualRE	0.22	0.38	0.27	0.27	0.44	0.34	0.29	0.62	0.40	<b>0.43</b>	0.76	0.54
REEnsemble	0.19	0.33	0.24	0.29	0.41	0.34	0.30	0.54	0.38	0.39	0.62	0.48
SelfTraining	0.21	<b>0.68</b>	0.31	0.21	0.70	0.33	0.25	<b>0.77</b>	0.37	0.31	0.85	0.45
$\text{SSL-RE}_{sdp}$ (1-Iter)	0.17	0.63	0.26	0.21	<b>0.71</b>	0.33	0.22	0.76	0.34	0.26	0.84	0.40
$\text{SSL-RE}_{sdp}$	<b>0.31</b>	0.51	<b>0.38</b>	<b>0.39</b>	0.65	<b>0.48</b>	<b>0.44</b>	0.71	<b>0.54</b>	<b>0.43</b>	0.78	<b>0.56</b>
<b>ChemProt</b>												
$\text{SUP-RE}_{sdp}$	0.095	0.34	0.15	0.13	0.40	0.20	0.21	0.54	0.30	0.31	0.76	0.44
LabelPropagation	0.12	0.50	0.20	0.10	0.42	0.19	0.14	0.56	0.24	0.21	<b>0.83</b>	0.33
DualRE	0.15	0.26	0.13	0.19	0.49	0.27	0.23	0.46	0.30	0.36	0.69	0.48
REEnsemble	0.032	0.13	0.051	0.066	0.25	0.11	0.11	0.40	0.18	0.22	0.7	0.33
SelfTraining	0.15	<b>0.54</b>	0.23	0.17	0.55	0.26	0.23	0.61	0.33	0.33	0.76	0.46
$\text{SSL-RE}_{sdp}$ (1-Iter)	0.13	0.39	0.20	0.17	0.52	0.26	0.24	<b>0.66</b>	0.35	0.30	0.76	0.43
$\text{SSL-RE}_{sdp}$	<b>0.29</b>	0.44	<b>0.35</b>	<b>0.40</b>	<b>0.56</b>	<b>0.46</b>	<b>0.47</b>	<b>0.66</b>	<b>0.54</b>	<b>0.46</b>	0.72	<b>0.56</b>
<b>PPI</b>												
$\text{SUP-RE}_{sdp}$	0.31	0.71	0.43	0.35	0.78	0.47	0.43	0.82	0.56	0.49	0.82	0.61
LabelPropagation	0.20	<b>0.91</b>	0.35	0.21	<b>0.99</b>	0.35	0.27	<b>0.95</b>	0.42	0.36	<b>0.88</b>	0.49
DualRE	0.38	0.52	0.28	0.33	0.78	0.46	0.41	0.64	0.47	0.61	0.64	<b>0.63</b>
REEnsemble	0.33	0.26	0.28	0.40	0.34	0.33	0.52	0.34	0.38	<b>0.67</b>	0.56	0.60
SelfTraining	0.38	0.74	0.43	0.34	0.78	0.47	0.38	0.87	0.53	0.48	0.84	0.61
$\text{SSL-RE}_{sdp}$ (1-Iter)	0.31	0.69	0.43	0.35	0.74	0.47	0.40	0.83	0.54	0.38	0.86	0.53
$\text{SSL-RE}_{sdp}$	<b>0.50</b>	0.54	<b>0.52</b>	<b>0.56</b>	0.66	<b>0.61</b>	<b>0.55</b>	0.77	<b>0.64</b>	0.39	0.84	0.53

$\text{SSL-RE}_{sdp}$  achieved consistent gain over Label Propagation, Self-Training, DualRE, RE Ensemble on all datasets and with different labeling budgets, except in PPI dataset trained on 500 budget where DualRE performed the best.

One can observe Self-Training and DualRE do not have stable performance due to reliance on the predictor to provide weak labels. For example, Self-Training outperforms DualRE in PPI dataset on [50, 100, 200] budgets, while underperforming DualRE on DDI and ChemProt occasionally. This provides evidence that predictor-based SSL models are sensitive to the performance of the RE model.

In addition, Label Propagation performed weaker than baselines which shows that its low quality of imputation damages the model’s performance.

It could be concluded that  $\text{SSL-RE}_{sdp}$  benefits from iterative augmentation, after comparing to  $\text{SSL-RE}_{sdp}(1\text{-ter})$ , which only uses one pass of label imputation. In addition, it improves the performance over the supervised baseline by a significant margin in all the experiments.

**5.5.2.1 Performance on Different Datasets.** The marginal gain of  $\text{SSL-RE}_{sdp}$  on PPI is smaller than on ChemProt and DDI in Table 5.4. This is because the size of PPI is  $4.2\times$  smaller than DDI and ChemProt. Therefore, the amount of unlabeled data may not be sufficient to identify the most similar neighbors. This can be observed on other baselines since they underperformed the supervised baseline on this dataset, except for DualRE trained on 500 budget.

**5.5.2.2 Performance as a Fraction of Labeled Data Size** Based on the results in Table 5.4,  $\text{SSL-RE}_{sdp}$  is the most advantageous when the labeled dataset is extremely small (around 100 - 500), which is common in Biomedical domain. In DDI,  $\text{SSL-RE}_{sdp}$  can reduce the need for labeled data by up to  $5\times$ ,  $\text{SUP-RE}_{sdp}$  achieves 0.46 F1 when trained on  $\mathcal{D}_L = 500$ , while  $\text{SSL-RE}_{sdp}[100]$  boosts  $\text{SUP-RE}_{sdp}[500]$  performance by 4% when using only  $\mathcal{D}_L = 100$ .

Similar outcome can be observed in ChemProt dataset.  $\text{SSL-RE}_{sdp}[100]$  is  $2\times$  more accurate than  $\text{SUP-RE}_{sdp}[200]$ , while using  $2\times$  less labeled data.  $\text{SSL-RE}_{sdp}$  is also more accurate than  $\text{SUP-RE}_{sdp}$  on PPI on 50, 100, and 200 budgets, reducing the labeling need by  $4\times$ , achieving 0.56 with  $\text{SUP-RE}_{sdp}[200]$  and 0.52 with  $\text{SSL-RE}_{sdp}[50]$ .

Overall, one can observe that  $\text{SSL-RE}_{sdp}$  is significantly beneficial when the cost of collecting labeled data is very high.

**5.5.2.3 Statistical Significance Test.** The t-test  $\text{test}^2$  for statistical significance has been used to find whether the difference between  $\text{SSL-RE}_{sdp}$  and other SSL baselines are

<sup>2</sup> Student’s t-test. <https://en.wikipedia.org/wiki/Student%27st-test>. Accessed 11 July 2023

Table 5.5. T-test analysis of SSL-RE<sub>sdp</sub> versus baselines.

	T-statistic	P-Value
LabelPropagation	9.5	3e-14
DualRE	4.8	1e-05
REEnsemble	6.8	3e-09
SelfTraining	4.9	6e-06

due random chance. Therefore, we define the null hypothesis as there is not a significant difference in the performance of SSL-RE<sub>sdp</sub> and other baselines. To this end, we use the final F1 scores from 3 independent runs across 4 labeling budgets to calculate p-value and t-statistics. We report the pvalue of our method compared to label propagation, self-training, RE-Ensemble, and dualRE in Table 5.5. The reported results reject the null hypothesis for all the baselines as they are all less than the significance level of 0.05, meaning our results are significantly better than baselines. This can be confirmed through t-statistic’s magnitude, since it is positive which indicates a higher difference between the average performance of SSL-RE<sub>sdp</sub> versus baselines and suggests stronger evidence against the null hypothesis.

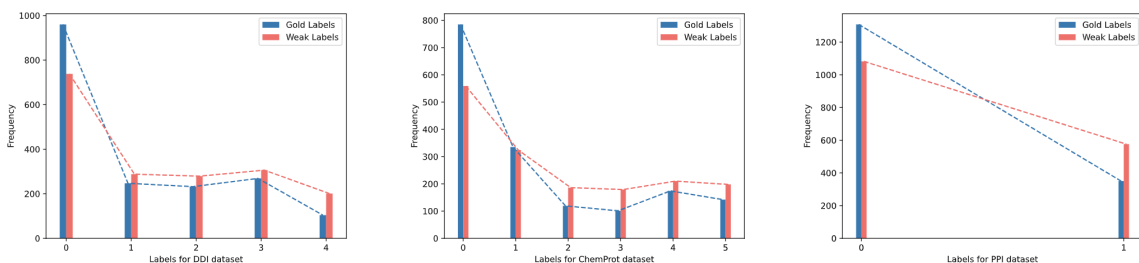


Figure 5.3. Comparing the distribution of imputed labels in the augmented examples (red bars) to their actual labels (blue bars) on DDI, ChemProt, and PPI dataset.

**5.5.2.4 Imputation Bias Analysis.** To validate SSL-RE<sub>sdp</sub> could prevent any imputation bias (e.g. a certain RE type is overpredicted) due to label noise and enables quality weak labels, in Figure 5.3, we represent gold label distribution with blue and weak label

distribution with green. Technically, we have the ground truth labels available for all imputed weak labels. From Figure 5.3, we observe that weak label distribution is close to the gold label distribution with less drift.

**5.5.2.5 Qualitative Analysis of SSL-RE<sub>sdp</sub> Versus Baselines.** Figure 5.4 demonstrates few examples of the actual prediction of baseline models vs SSL-RE<sub>sdp</sub>. All models are trained using SDP finetuning. SDP tokens used in finetuning are highlighted as green. In the first four examples, SSL-RE<sub>sdp</sub> can accurately captures the gold relations between entities, while in the last example self-training and RE-Ensemble performed better.

Sentence	Gold	LP	ST	DR	RE	SDP <sub>ssl</sub>
We propose that @PROTEIN\$ interacts with the @PROTEIN\$ in a fashion that is different from that used by p52Shc.	True	False	False	False	False	True
@PROTEIN\$ induces chemotaxis and adhesion by interacting with CCR1 and @PROTEIN\$.	False	False	False	False	False	False
@CHEMICAL\$ prevented the PT in mitochondria directly and also indirectly through induction of antiapoptotic BC6OTHER and a @GENE\$ , BC6OTHER ( BC6OTHER ).	CPR:3	False	CPR:3	False	CPR:4	CPR:3
@GENE\$ continues to be a critical target for BC6OTHER and its prodrugs, primarily because this enzyme is essential for the synthesis of @CHEMICAL\$ , a precursor for DNA synthesis.	CPR:9	False	False	False	False	CPR:9
@CHEMICAL\$ is an inhibitor of @GENE\$ which is able to block the absorption of 30% of ingested fat.	CPR:4	False	CPR:4	False	CPR:4	False

Figure 5.4. Qualitative Analysis of SSL-RE<sub>sdp</sub> vs baselines on PPI and ChemProt datasets. LP, ST, DR, RE are denoted as Label Propagation, Self-Training, Dual-RE, and RE-Ensemble.

### 5.5.3 *In-Context Learning*

The aim of this experiment is to assess the effect of utilizing Shortest Dependency Path (SDP) representation to boost the accuracy of Relation Extraction (RE) within the in-context-learning framework. To this end, we furnish the GPT-3 model with task-specific instructions and a few examples that illustrate the task at hand.

Recent research Liu et al. (2021); Rubin et al. (2021) indicates that dynamically selecting in-context examples for each test instance, rather than employing a fixed set of in-context examples, results in notable improvements in GPT-3’s in-context learning. Taking inspiration from the approach outlined in Liu et al. (2021), we implement a k-nearest neighbor (kNN) retrieval module to identify the most closely related examples from our constrained training dataset to serve as the in-context prompts for each test instance. During this process, we use the SDP representation as the basis for calculating the distance metric, which in turn determines the similarity between the test and training instances.

In our experiments, we allocate a training budget of 50 and, in each test, we contrast the efficacy of SDP-based nearest neighbor retrieval with random and fixed prompting. In the random prompting scenario, we arbitrarily select in-context examples from the training dataset for every test instance, while in the fixed prompt setting, we maintain a consistent set of examples across all test sets. We employ stratified sampling to ensure each relation type is represented in the prompt along with the task instruction. To carry out this task, we leverage the highly potent GPT-3 DaVinci engine. However, due to cost considerations associated with using GPT-3, we restrict our test set to a subsample of 200 examples for each experiment. For both fixed and random prompts, we repeat the experiments three times (keeping the test set constant but varying the in-context examples) to establish the reliability of our results.

As depicted in Table 5.6, the inclusion of SDP-based nearest neighbor retrieval in Drug-Drug Interaction (DDI) led to a considerable improvement in performance for both fixed

Table 5.6. Using SDP to retrieve NN in few-shot experiments versus random and fixed example selection in prompts in in-context-learning.  $SDP_{nn}$  indicates using SDP to retrieve nearest neighbors.  $SSL-RE_{sdp}$  indicates the semi-supervised performance on 50 training budget and tested on the same test set (200 examples.)

	DDI			ChemProt			PPI		
	P	R	F1	P	R	F1	P	R	F1
$SSL-RE_{sdp}$	0.63	0.50	0.56	0.69	0.43	0.53	0.81	0.62	0.71
GPT3 (fixed)	0.45	0.37	0.40	<b>0.72</b>	0.40	0.51	<b>0.61</b>	<b>0.80</b>	<b>0.69</b>
GPT3 (random)	0.45	0.41	0.43	0.69	0.36	0.47	0.60	0.78	0.68
GPT3 ( $SDP_{nn}$ )	<b>0.51</b>	<b>0.50</b>	<b>0.50</b>	0.71	<b>0.41</b>	<b>0.52</b>	<b>0.61</b>	<b>0.80</b>	<b>0.69</b>

and random prompts. A modest positive effect was observed for ChemProt, with an approximate increase of 1.9% in performance. However, no discernible improvement was recorded for Protein-Protein Interaction (PPI).

#### 5.5.4 Ablation Study

**5.5.4.1 Choice of Representation On Augmentation Module.** We investigate the impact of SDP on the label imputation. To this end, we performed experiments on different sequence representations to compute distance metric, and thereafter used the imputed labels to train the RE model. Note that we keep  $SUP-RE_{sdp}$  architecture for the RE model training, and only changed the representation in the imputation. We use the following representations to extract from the last hidden state of the encoder  $\mathcal{Q}$ .

- CLS (L): is an aggregate representation of all the tokens in a sentence
- ent-avg (L): is the average embedding of the entities in a sentence
- ent-sdp-avg (L): is the average embedding of the entities and SDP tokens
- ENT (2L): is concatenation of the embeddings of two entities in a sentence
- ent-words-between (3L): is concatenation of the embeddings of the two entities along with the average representation of all the words between two entities
- $SDP_{rep}$  (3L): is our proposed representation in equation 5.5

As shown in Table 5.7,  $SDP_{rep}$  representation results in overall better F1 score compared to other representations. By comparing the average performance of all representations across all datasets, we observed that  $SDP_{rep}$  ranked highest achieving 0.39 average F1 score, ent-sdp-avg ranked second with 0.38 average F1, and CLS ranked lowest with 0.35 average F1.

The representation ent-words-between achieved second to the last (0.37 average F1), meaning adding unnecessary context does not help to find high-quality neighbor search.

Table 5.7. Impact of representation choice in augmentation module, and the resulting performance of RE model. We experimented with 200 labeling examples.

	Dim	DDI	ChemProt	PPI
CLS	L	0.30	0.24	0.51
ent-avg	L	0.30	0.33	0.52
ent-sdp-avg	L	0.32	0.31	0.53
ENT	2L	0.30	0.33	0.51
ent-words-between	3L	0.32	0.28	0.52
$SDP_{rep}$	3L	<b>0.32</b>	<b>0.33</b>	<b>0.54</b>

**5.5.4.2 Effectiveness of Soft Labels.** To better understand the impact of soft label assignment in weak label imputation, Table 5.8 reports the performance against hard label assignment, where we only take the label with highest probability during training. We could see that soft labels improve the performance on DDI and ChemProt datasets by 13% and 8% in F1. There is no improvement over PPI dataset.

Table 5.8. Effectiveness of soft label assignment in three datasets using 200 training data.

	DDI	ChemProt	PPI
SSL- $RE_{sdp}$ (hard label)	0.32	0.32	<b>0.52</b>
SSL- $RE_{sdp}$ (soft label)	<b>0.37</b>	<b>0.35</b>	0.51

## 5.6 Discussion and Limitation

Our study demonstrates the SDP’s linear scalability which is a critical factor for practical large-scale applications. In practical testing, SDP generation took only 8.86 seconds for 500 samples, while scaling to larger datasets, such as 2000 samples, necessitated a proportional increase in computation time to 34.05 seconds. This efficient preprocessing enables the model’s use in extensive literary corpora, such as PubMed abstracts, without imposing significant computational delays.

Our findings suggest that the integration of SDP with nearest neighbor enriches the model with nuanced syntactic and semantic context while carefully imputing pseudo labels. However, as dataset sizes grow, the method’s relative benefit may diminish due to stronger inherent patterns within the data. Nevertheless our approach offers a pragmatic and feasible solution for initial analyses, beneficial for users needing immediate insights without the complexity of larger models.

Moreover, the SDP representation can seamlessly augment the capabilities of existing off-the-shelf RE models, thereby enhancing their accuracy and reliability for comprehensive analysis.

We acknowledge certain limitations in our methodology. One limitation of our work is that we assume that unlabeled and labeled data are sampled from the same distribution. If the sampling of labeled data is biased, our label imputation approach may not work that well.

Second, our approach depends on the availability of a good dependency parser. This is a limitation if the proposed approach is used on rare languages or in very specialized domains. Third, all three of our datasets had a relatively small number of clearly delineated relation types. It would be important for future work to exploit the effectiveness of the proposed approach on data with a much larger number of relation types.

Fourth, our experiments were performed using the BERT encoder. While it is one of the first strong LLM models, we have recently witnessed the emergence of much stronger models such as GPT-4. It remains an open question if SDP representation could be helpful to those newer LLMs. There are two reasons we did not use GPT-4 or comparable models. First, most of those models are proprietary and inaccessible to researchers. The open-sourced versions are typically much weaker for multiple reasons. In addition, the state-of-the-art LLMs are also extremely large, and our lab did not have sufficient computational resources to support experimenting with those models.

## **5.7 Conclusion**

This study demonstrates the utility of Shortest Dependency Path (SDP) representations in supervised, semi-supervised, and in-context learning for low-resource biomedical relation extraction (RE). We introduced an innovative SDP-based representation, which we employed to compute the distance metric between RE instances. In addition, we proposed a new semi-supervised learning (SSL) algorithm tailored for biomedical RE. Comprehensive experimental assessments on three biomedical text datasets substantiate the effectiveness of SDP representation. Importantly, our proposed approaches are not tied to a specific neural network architecture and can be seamlessly integrated as a wrapper around existing and future RE models.

## CHAPTER 6

### AUTOMATED NARRATIVE SCORING USING LARGE LANGUAGE MODELS

#### 6.1 Introduction

A narrative is a monologic telling or retelling of causally and temporally related events (i.e., a story), such as a problem to solve, emotional responses to the problem, attempts to solve the problem, and the consequence of those attempts Stein & Glenn (1975); Peterson & McCabe (1983). A sizable literature documents strong associations between narrative abilities and academic achievement Bishop & Edmundson (1987); Dickinson & McCabe (2001). For example, children's early oral narrative abilities predict later reading comprehension Catts et al. (2002); Snow et al. (2007) and writing performance Scott & Windsor (2000); Kim et al. (2015).

Because of their academic value, teachers and school-based speech-language pathologists play a key role in helping children develop their narrative skills. Analyzing child-produced narrative samples is an essential component of language and academic assessment. To analyze the quality of a narrative, it is common to examine the inclusion and clarity of discourse components, often referred to as narrative discourse elements. Al-mubark et al. (2023) found that narrative discourse was the only consistent and significant predictor of disability among students in grades K-3. At this macroorganizational level, educators review transcribed oral language samples for each of the canonical elements of character, setting, problem, feeling, plan, attempt, consequence, and resolution Stein & Glenn (1975) and rate their presence and clarity. Educators can enhance their analysis of narrative complexity by evaluating the usage of advanced vocabulary, the prevalence of subordinate clauses, and the intricacies of syntax. This detailed degree of narrative analysis is often termed as microstructural analysis or literate language assessment Greenhalgh & Strong (2001).

Despite its importance for educational achievement, rating a child's narrative production is timeconsuming and laborintensive. Educators must first learn what language features to look for and how to judge their complexity, which requires a deep knowledge of language structures and language development. Once the knowledge is acquired, educators must follow some type of standardized scoring system to determine the quality of the narrative produced. The work and time required to do this is sometimes beyond what educators can accommodate in their busy schedules.

Over the last few decades, there has been an increasing interest in using automated systems by leveraging the latest advances in artificial intelligence and computational linguistics. The automatic scoring of narrative productivity (e.g., total word count, number of communication units) and linguistic complexity (e.g., use of subordination and average utterance length) has a long history, including Childhood Language Analysis [CLAN] MacWhinney & Snow (1985), Systematic Analysis of Language Samples (SALT) Miller et al. (2016), and Literate Language Use in Narrative Assessment (LLUNA) Fox et al. (2022). These methods largely rely on metrics derived from word counts. Rulebased systems, such as the one proposed by Hsu & Thompson (2018), employ manual coding along with lexical and morphological variables. Hassanali et al. (2012) refined automated scoring by incorporating CohMetrix features, which analyze readability, the complexity of the situation model, word choice, syntax, and coherence. Random forest classifiers were employed to predict narrative quality by combining expert-crafted linguistic features in Somasundaran et al. (2015). Rulebased systems aim to analyze patterns in speech production across different levels of microsyntax, including utterance, sentence, lexical, morphological, and verb argument structure.

Deep learning techniques have a potential to further improve quality of automatic assessments Ranade et al. (2022). Transformer-based models, such as BERT Devlin et al. (2018), were evaluated recently Jones et al. (2019); Fernandez et al. (2022) and they have shown superior performance in scoring narratives based on the established rubrics such as

the Test of Narrative Language (TNL) Gillam & Pearson (2004a) and Monitoring Indicators of Scholarly Language (MISL) Gillam et al. (2017). Jones et al. (2019) showed that the advantage of BERT and similar models lies in their ability to process both handcrafted features and raw text, offering a more nuanced view at the narrative quality. Large language models (LLMs) such as GPT3 and Instruct-GPT Brown et al. (2020); Ouyang et al. (2022) opened new frontiers in natural language processing (NLP), demonstrating remarkable capabilities across a range of tasks, including text classification Sun et al. (2023). Researchers have already started employing GPT3 for essay grading under diverse evaluation criteria by supplying instructional prompts to the model Mizumoto & Eguchi (2023b). However, there is no comprehensive study yet that evaluates LLMs on the task of narrative analysis. This paper aims at addressing this research gap.

In this study, we aim to evaluate the performance of LLMs in assessing narrative content. By focusing on the accuracy of these models, we seek to identify their strengths and limitations in various storytelling contexts. The core of our investigation revolves around the following primary research questions:

**Q1-1. Accuracy in Discourse Analysis:** How accurately do LLMs assess different narrative discourse elements? This question aims to understand the models' ability to interpret key narrative components.

**Q1-2. Comparison with Human Expertise:** How does the performance of LLMs in rating narrative quality compare to that of human experts? We will specifically look at the comparison between LLM accuracy and human interrater reliability.

**Q1-3. Ability to Rate OutofSample Narratives:** Is there a gap in LLM accuracy between in-sample and out-of-sample stories? This examines the models' flexibility and generalizability in various storytelling scenarios.

To deepen our understanding of the factors that influence LLM accuracy, we also explore following secondary research questions:

**Q2-1.** Impact of Training Data Size: How does the amount of data available to train an LLM affect its accuracy in narrative assessment? This question seeks to explore the relationship between training data size and model accuracy.

**Q2-2.** Optimal Model Choice and Use Case Scenario: Among the current LLMs, which model and application scenario demonstrate the greatest ability for automatic narrative assessment? Here, we aim to identify specific models and use cases that stand out in their assessment capabilities.

Through this comprehensive examination, we aim to obtain insights that will enhance the deployment and development of LLMs in narrative analysis tasks. By addressing these questions, we hope to contribute valuable knowledge to the areas of automated story understanding and evaluation.

## **6.2 Materials and methods**

### **6.2.1 Narrative Data Sets**

LLMs were evaluated on two data sets. The first is the Academic Language of Primary Students data set Enayati et al. (2024) and the second data set is the Alien Story data set Jones et al. (2019). Each dataset is described in the following subsections.

**6.2.1.1 Academic Language of Primary Students (ALPS) Data** This data set comprises 3,484 narratives produced by school-aged students in kindergarten, first, second, and third grades. Samples were elicited using a standardized protocol employing either a retelling or generation task related to a provided set of pictures. There were nine sets of three pictures (referred to as story types). Researchers would randomly pick a picture set and display the three related pictures and either model a grade-appropriate story and ask the student to retell it (retell condition) or ask the student to generate a story about the

Table 6.1. Definition of rubrics and their score ranges for each dataset

Rubrics	Definition	Alien's Data	ALPS Data
Character (Char)	The who or what in the story acting as the agent	0-3	0-3
Setting (Sett)	The time and/or place the story or episode takes place	0-3	0-3
Initiating Event \ Problem (Prob)	An event or problem that causes the story to take-off	0-3	0-4
Plan (Plan)	The idea the character has to fix the problem in the story	0-3	NA
Action (Act)	The action taken by the character in response to the initiating event	0-3	NA
Plan & Attempt (Att)	The plan and action that the main character does to solve the problem	NA	0-4
Consequence (Con)	A casually linked event following the main character's action	0-3	0-4
Ending (End)	The action the main character did at the end of the story	NA	0-2
Emotion (Emo)	The main character's feeling about the problem	NA	0-3

pictures without modeling (generation condition). The same number of narrative samples were collected for each condition.

Researchers' audio recorded students' retold or generated stories, transcribed the samples, and then scored them using the Narrative Language Measures (NLM) Flowchart Petersen & Spencer (2012). The NLM Flowchart is a scoring rubric with a decision-tree format that allows for quick scoring of Discourse Complexity and Sentence Complexity. Only the scores yielded from the Discourse Complexity scales were used in this study. The rubrics align with oral and written academic language expectations set forth by the Common Core State Standards (CCSS; National Governors Association Center for Best Practices & of Chief State School Officers (2010)) and based on Stein & Glenn (1975) story grammar schema. When scoring each narrative, researchers rated the inclusion, clarity, and completeness of the following elements: Character, Setting, Problem, PlanAttempt, Consequence, Ending, and Emotion. Although most discourse elements were given a score ranging between 0 (not present) and 3 (complete and clear), samples that included more than one Problem, PlanAttempt, and/or Consequence were awarded an additional point for each additional component (see Table 6.1 for a description of each element). For more information about the scope of ALPS dataset, please refer to Spencer & Staff (2023).

**6.2.1.2 Alien Story Data** The Alien Story corpus consists of 414 narratives elicited from students ranging from 5 to 11-years-old. This dataset had previously been utilized by Jones et al. (2019), whose method we replicate in our study based on their work with this dataset. The narratives were generated in response to prompts from the Test of Narrative Language-2 (TNL) Gillam & Pearson (2004a). The story corresponded to a single image depicting an alien family landing in the park. The narratives produced by children were audio recorded and transcribed according to SALT conventions Miller et al. (2016). The narratives were scored using a scoring rubric, Monitoring Indicators of Scholarly Language (MISL) Gillam et al. (2017). The MISL includes subscales for macrostructure (i.e., discourse elements) and microstructure (i.e., linguistic complexity). As per Jones et al. (2019), six primary macrostructural elements were considered, each receiving scores ranging from 0 to 3: Character, Setting, Initiating Event, Plan, Action, and Consequence. The description about each element followed by their score ranges is shown in Table 6.1.

## **6.2.2 Models for Narrative Assessment**

The task involves the use of an LLM to predict integer scores corresponding to each discourse elements for a given narrative. Formally, let  $s$  be a narrative and  $r = \{r_1, r_2, \dots, r_m\}$  be a list of assessment scores for  $m$  discourse elements. The task is formulated as a prediction problem, where the objective is to learn a function,  $f(s) = r$ , that automatically predicts scores  $r$  accurately.

The LLM model of emphasis for this research is the Generative Pre-trained Transformer (GPT) Brown et al. (2020), a model that stands at the forefront of language modeling due to its sophisticated capabilities Han et al. (2021); Dinh et al. (2022). GPT models are pre-trained on diverse and expansive datasets compiled from a wide swath of the internet, including books, articles, and websites, encompassing virtually every domain of knowledge available to the public Brown et al. (2020). This pre-training involves learning patterns of language, factual information, and even stylistic nuances from hundreds of gigabytes of text

data. For instance, GPT3 is a version of GPT model that has different versions, from the most powerful and expensive version, Davinci, with 175 billion parameters, to the smallest yet cheapest to use engine, Ada, with 40 million parameters. Larger GPT3 models are known to generate coherent and human-like text.

GPT models can be used to solve NLP tasks in two ways: in-context learning and fine-tuning. In-context learning provides the GPT model with a text input, known as a “prompt”, instructing the model to generate specific content. In-context learning leverages model’s pre-trained knowledge on diverse data set to produce contextually appropriate responses based on the provided prompt. In-context learning does not update any weights in the GPT model. Unlike in-context learning, fine-tuning is a more specialized approach that adjusts the model’s weights to produce the desired response. It updates the weights of the pre-trained GPT model using task-specific training data consisting of pairs of inputs (e.g., stories) and responses (e.g., story assessments).

The choice of whether to fine-tune or use in-context learning depends to large extent on the size of the training data set. In-context learning is more appropriate when the training data is small or even when training data does not exist. Alternatively, GPT fine-tuning is desirable when the training data set is sufficiently large.

**6.2.2.1 GPT3 Fine-tuning** GPT3 Fine-tuning The fine-tuning approach was explored and tested in this study. Fine-tuning updates the weights of a pre-trained model using task-specific training data. To fine-tune GPT3, we used narrative texts that were rated by educators. There are two ways to fine-tune GPT: I) train a separate GPT model for each discourse element; II) train a single GPT model that can predict the scores for all discourse elements. We chose to do the latter one as this approach reduces the training cost by a factor  $m$ , where  $m$  is the number discourse elements. During training phase, the model is provided with a narrative text as input and is trained to predict the scores for all discourse elements. The responses were formatted in a specific template form for each element. For

example, if the score on Character was 2, we generated the following desired response: “Character receives 2 points out of 3.” In Figure 6.1, we illustrate the workflow of our fine-tuning and show the format of input and response text. Our preliminary experiments show that generating response in the proposed manner is better than using pure scores as outputs. During testing phase, we applied the fine-tuned GPT3 model on testing data, which has the same format as the training data. The model was then tasked to generate the text-based scores for all elements. We then used regular expressions to extract the numerical score from these textual responses to evaluate model’s accuracy.

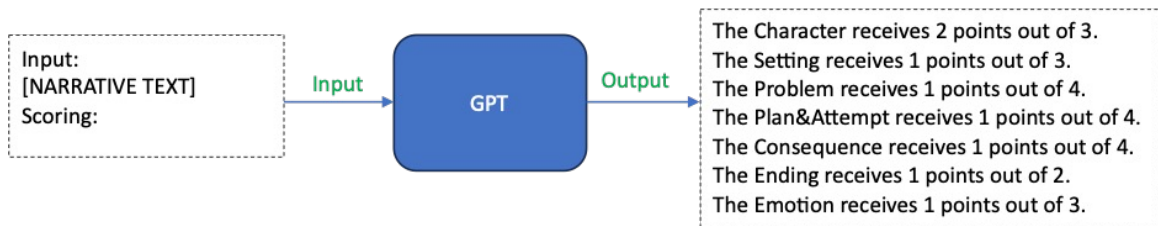


Figure 6.1. GPT3 fine-tuning illustration. Illustration of the input-output training example created to fine-tune GPT3

**6.2.2.2 GPT3 In-Context Learning** GPT3 In-cotext As we explained, in-context learning is a method to let LLMs adapt to new tasks based on the information provided in the input prompt, without further training. The prompt is the important cue to guide the model generating responses aligned with the task and users’ expectations. The careful design of the prompt is crucial because the model is sensitive to the input prompt. Among several approaches in prompt engineering Liu et al. (2023), we chose the chain-of-thought (CoT) Wei et al. (2022) technique where GPT is instructed to explain how to arrive at the answer before providing the answer. To construct the prompt, we follow the CoT instruction with a few examples of desired assessments of narratives. At the end of the

prompt, we input a narrative we wish to score and expect GPT to provide scores with justifications.

In Figure 6.2, we show an example of our designed prompt. In particular, we crafted prompts that incorporated task instructions, six example narratives with justifications for why each element was scored the way it was, and a numerical rating for each discourse element.

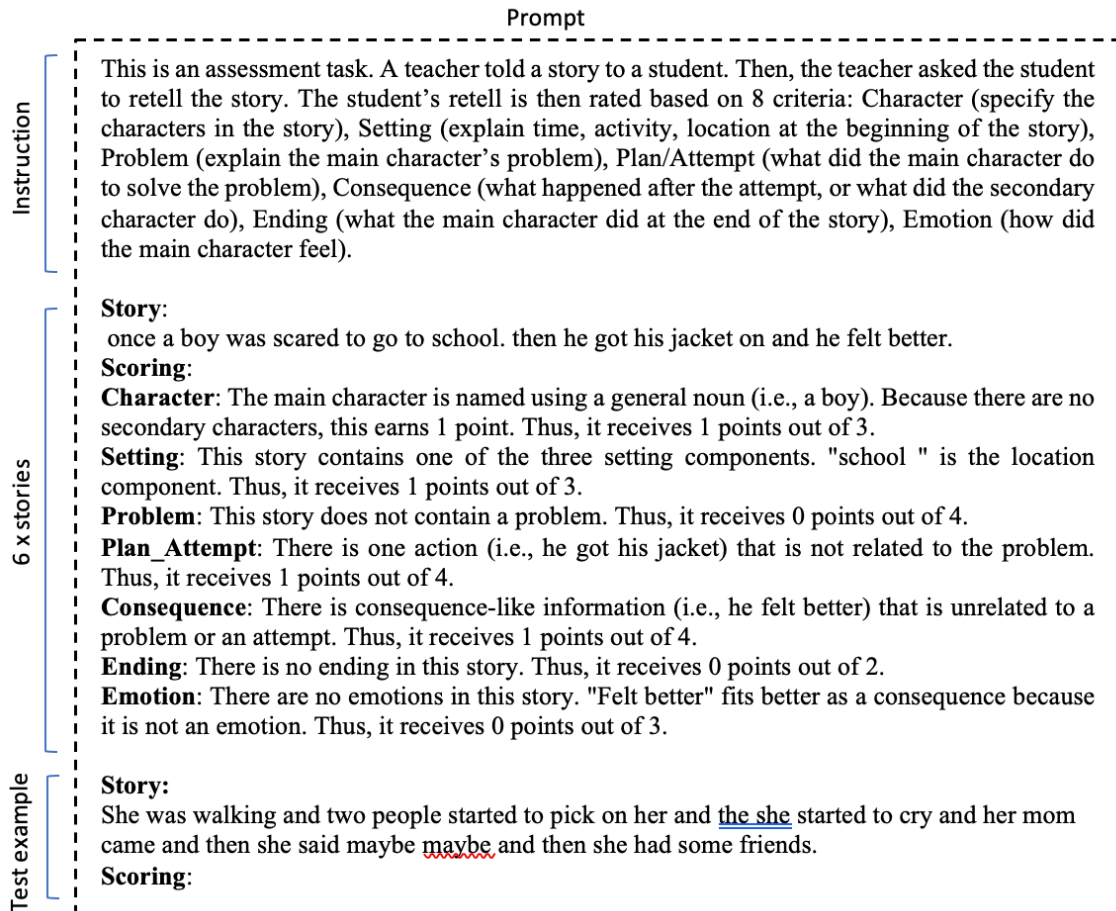


Figure 6.2. GPT in-context-learning prompt. Example of the prompt used for in-context learning. 1 story enriched with reasoning are provided as the task demonstration followed by the text example to be rated by GPT.

Since GPT is a generative model, it always generates text. Therefore, we have to post-process the generated response to extract the predicted scores and justifications. We extracted the numerical scores from the textual responses using regular expressions.

**6.2.2.3 Baseline Models** We compared GPT models with two baselines: Decision Tree Sammut & Webb (2011) and BERT Devlin et al. (2018). Decision tree is a classical machine learning approach, while BERT is a language model that was the state of the art prior to introduction of GPT. BERT was a model of choice in the previous work on evaluation of LLMs on narrative assessment by Jones et al. (2019). Both approaches rely on the standard machine learning process: to train a model, they take the raw text (narrative) as input, and learn to map input to output (narrative discourse elements). To test the model, given the trained model, the narrative text is used as input and the model predicts the output, which is the score corresponding to discourse elements. The two types of baselines are described next.

**Baseline DT (Decision Tree):** Decision tree Sammut & Webb (2011) is a popular supervised learning ML algorithm that creates a tree-structured model that makes a prediction based on a series of yes/no questions and can be used for classification and regression tasks. In our study, we specifically employed the decision tree algorithm for a regression task. The objective was to predict the score for an element based on the length of each narrative (number of words split by white-space), which, in this method, serves as a single input variable for the model. The justification for using decision tree with this single variable is our observation that longer narratives are more likely to be highly assessed than very short ones. A successful deep learning model, such as BERT or GPT, should achieve significantly higher accuracy than this model. Thus, the decision tree serves as a lower bound on achievable accuracy. Since the decision tree can provide a decimal as output, we rounded its prediction to the nearest score.

**BERT:** Bidirectional Encoder Representations from Transformers (BERT) Devlin et al. (2018) achieves high accuracy on a variety of tasks such as sentence classification, question answering, and sentence tagging. BERT takes a raw text as input and transforms each word in the input to numerical vector representation (e.g., 768-dimensional embedding vector). One of the key features of BERT is its ability to learn contextual word embeddings. The

bidirectional nature of BERT allows it to consider the entire input sequence in both forward and backward directions, capturing rich contextual information. The BERT model is pre-trained on a large corpus of text, and the learned representations can be fine-tuned on a wide range of NLP tasks. Jones et al. (2019) adapted BERT to predict MISL flowchart scores of narratives from the Alien dataset. In this study, we consider BERT as a competitive baseline for GPT. A narrative is provided as input to the BERT, which creates vector representation of the narrative (denoted as CLS token). This vector representation is then linearly transformed to provide numerical assessment of the narrative. We fine-tuned the BERT model for a regression task to minimize mean squared error (MSE) of the assessment score. The predicted numerical score is then rounded to the nearest integer score and compared with the actual score to compute accuracy. Separate BERT models were trained for each discourse element.

**6.2.2.4 Hyperparameter Tuning and Optimization** Hyperparameter tuning refers to the process of selecting the parameters of a machine learning algorithm that need to be defined before model training. It is a critical step when applying most types of machine learning algorithms.

**Hyperparameters for GPT3 fine-tuning:** We opted to use GPT3 Ada engine for fine-tuning due to its cost-effectiveness. According to OpenAI pricing <sup>1</sup>, fine-tuning with Ada is 75 times cheaper than Davinci (stronger engine). Also, our limited preliminary results showed that fine-tuned GPT3 Ada achieves similar accuracy to fine-tuned GPT3 Davinci. The hyperparameters for fine-tuning GPT3 were mostly set to their default values according to OpenAI manual <sup>2</sup>. The maximum number of training epochs was set to 15 and fine-tuning was allowed to be terminated early when the accuracy on validation set that was randomly selected from training data stopped improving. We allowed GPT3 to generate up

---

<sup>1</sup> <https://openai.com/pricing>

<sup>2</sup> <https://platform.openai.com/docs/guides/fine-tuning/hyperparameters>

to maximum of 100 tokens. The temperature parameter that controls the randomness of the generated output was set to 0, meaning that GPT always generated the most likely word. We used the stop token delimiter ‘##’ to indicate the end of a sentence. It served as a signal to the model that it should stop generating text.

**Hyperparameters for In-Context Learning:** We used GPT3.5 and GPT4 for our in-context learning experiments. As we described in Section GPT3 In-cotext, in-context learning does not require ample amount of training data. In our experiment, we used only six scored narratives with justifications in the prompt, as shown in Figure 6.2. We used a larger value of 512 for the output length since this approach generates justifications in addition to scores. Also, to improve the models’ creativity in reasoning, we used the temperature 0.2.

**Hyperparameters for Baselines:** For decision tree, we used the implementation in scikit-learn python package <sup>3</sup> and used default hyperparameters values. For BERT baseline, we followed the methodology and used the same hyperparameters as in Jones et al. (2019). We opted for the BERT-base-uncased pretrained model for fine-tuning in the regression task. Across all elements, the learning rate was set to 5e-6, the batch size to 16, and maximum number of training epochs to 30. Throughout the fine-tuning process, we preserved the best-performing model based on its performance on the validation set, ensuring an accurate evaluation on the test set.

### 6.2.3 *Evaluation Metric*

In the evaluation of our predictive models, we followed Jones et al. (2019) who used Quadrative Weighted Kappa (QWK) to evaluate the agreement between automatic scoring and ratings from human experts. It does so by analyzing a confusion matrix and calculating the weighted difference between predicted and actual scores. QWK has the advantage over other classification metrics as it assigns higher weights to agreements that are closer to perfect agreement, and lower weights to agreements that are further from perfect agree-

---

<sup>3</sup> <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html>

ment. The score ranges between zero and one (the higher the better) and any score above 0.6 is treated as a strong agreement in the literature Dikli (2006). In our analysis, QWK for both automatic scores and scores from a secondary human coder were compared with the primary coder. This allowed us to both calculate the accuracy of the machine learning models and to compare with the human interrater reliability.

#### **6.2.4 Experimental Design**

To answer our research questions, we divided the ALPS dataset into several subsets. The dataset comprises nine distinct story types, which we divided into two groups for analysis: 1) In-sample set: This consists of narratives from six story types, including 1,686 narratives for training and 573 for testing. We use term “in-sample” because these story types were shown to the model during training. 2) Out-of-sample set: This includes 1,039 narratives from the remaining three story types used exclusively for testing. We refer to this set as “out-of-sample” since these story types were not shown to the model during training.

In total, our study involved training on 1,686 in-sample narratives, and 1,612 (573 in sample and 1,039 out-of-sample) for testing, as depicted in Figure 6.3. ALPS narratives were rated by one of 10 trained human coders, which we refer to as primary human coders. The assessments by the primary coder were treated as gold standard scores in our experiments. For a subset of ALPS narratives, assessments from another human coder were available. We made sure that all those narratives were reserved as the test data set, unseen during training. In particular, there were 573 narratives from the in-sample set and 250 examples from the out-of-sample set that had been assessed by the secondary human coder. This enabled us to calculate inter-rater reliability between human coders, offering a basis to measure the reliability of the model generated scores. Moreover, evaluation on the out-of-sample set is important to understand the generalizability of the models on new story types.

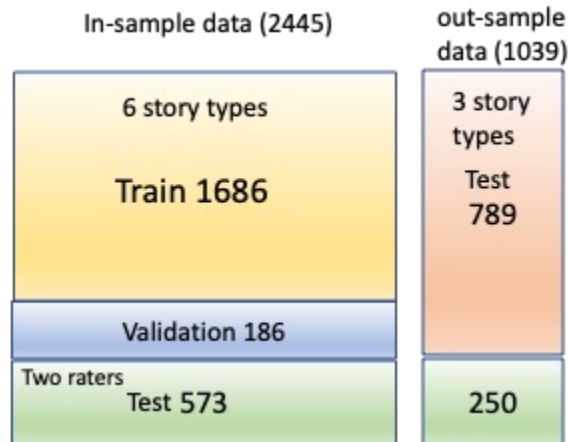


Figure 6.3. Data splits for ALPS data. The green portion of the figure (823 stories) were stories rated by two raters. Panels on left correspond to six story types (in-sample) that were seen in training data. Panels on right correspond to three story types (out-of-sample) that were not seen in training data.

On the Alien dataset we follow Jones et al. (2019) to maintain consistency and comparability, we used the same training and test split, involving 331 training and 83 testing narratives, respectively. Since there was only one story type rated by a single expert coder, we were not able to measure the interrater reliability on this dataset.

### 6.3 Results

Our main goal was to evaluate the accuracy of GPT models on ALPS and Alien Story narrative datasets and compare it with BERT model Jones et al. (2019), DT baseline, and with the human interrater reliability. In Table 6.2 we show QWK scores of DT, BERT, and GPT3 Ada models on Alien Story data. All three listed models were trained on 331 examples and tested on 83 examples from Alien data. The columns list QWK scores for each of the six discourse elements explained in Table 6.1 and the averaged QWK score. The results show that average QWK score of GPT3 (60.95) was substantially higher than BERT’s average score (51.51). The average QWK of DT was the lowest (30.91). Looking at the individual discourse elements, we can observe some variance in the results that could

Table 6.2. QWK accuracies of different models on 83 test stories from Alien Story Data.

	Char	Sett	Prob	Plan	Act	Con	Avg QWK
Baseline DT	28.78	22.19	41.43	7.56	55.50	30.0	30.91
BERT	<b>91.80</b>	30.30	50.30	37.70	<b>54.9</b>	44.1	51.51
GPT3	91.50	<b>51.19</b>	<b>57.84</b>	<b>62.68</b>	50.76	<b>51.76</b>	<b>60.95</b>

be attributed to the relatively small test set of 83 examples. It seems that Character is easier to predict than other discourse elements.

We observe that BERT results in Table 6.2 are not consistent with the same results reported in Table 2 in Jones et al. (2019). Our reported accuracies are substantially lower. Our investigation of BERT code provided by Jones et al. (2019) revealed that there was an overlap between testing and training data in this previous work. Testing on training data is known to result in overly optimistic accuracies and the standard practice in machine learning is to test on examples not seen during training. Our reported accuracies are consistent with the standard evaluation practice in machine learning.

On the ALPS data, we evaluated the model accuracy on the whole test set (Table 6.3) and several of its subsets (Table 6.4) as defined in Figure 6.3. In Table 6.3, the columns show the 7 discourse elements of ALPS data, explained in Table 6.1, and the average QWK scores. The rows correspond to QWK scores of different models tested on 1,612 examples, which covers the entire test set. We show the results of GPT3, BERT, and DT models trained on the full training data set of 1,686 as well as the results of GPT3 and BERT models trained on the 10% of the training set (denoted as “\*\*\*”). Our goal was to measure the relationship between the training data size and the model accuracy. According to Table 6.3, the results show that the average QWK of GPT3 generated scores on all the ALPS discourse elements (69.12) is higher than BERT(64.58) and substantially higher than DT (24.33). The QWK scores of GPT3 across different elements range between 53.53 and 86.21, with Character the most easily predictable element and Ending the hardest to predict.

Table 6.3. QWK accuracies of different models on 1,612 ALPS test stories. Numbers in parenthesis next to model names are the number of stories used in training.

	Char	Sett	Prob	Att	Con	End	Emo	Avg QWK
Baseline DT (1,686)	20.20	20.73	17.80	35.34	32.26	19.69	14.57	24.33
BERT** (168)	52.90	37.50	20.70	24.80	12.10	29.90	47.70	29.65
GPT3** (168)	83.46	54.16	16.87	51.52	37.48	36.03	73.84	46.58
BERT (1,686)	<b>88.10</b>	75.30	62.00	57.70	56.20	48.20	81.70	64.58
GPT3 <sub>Ada</sub> (1,686)	86.21	<b>78.20</b>	<b>68.21</b>	<b>64.70</b>	<b>63.90</b>	<b>53.53</b>	<b>82.91</b>	<b>69.12</b>

Table 6.4. Comparison of finetuning different GPT3 models versus in-context learning with GPT3.5 and GPT4. All models are tested on the same 10% of the randomly selected two-rated test ALPS stories (82 examples).

	Char	Sett	Prob	Att	Con	end	emo	Avg QWK
IRR baseline	95.80	78.89	82.45	80.73	70.94	65.19	79.92	79.13
Baseline DT(1686)	23.28	30.73	31.94	57.83	53.71	22.56	8.02	32.58
BERT** (168)	64.18	51.16	20.30	28.36	15.81	25.92	27.25	32.28
GPT3 <sub>Ada</sub> * *(168)	88.27	55.25	26.26	71.04	60.99	43.56	72.39	59.68
GPT3 <sub>Babbage</sub> ** (168)	78.81	56.88	27.26	65.61	50.55	44.67	65.18	55.56
GPT3 <sub>Curie</sub> ** (168)	83.40	66.82	29.88	79.67	66.94	52.03	72.30	64.44
GPT3 <sub>Davinci</sub> ** (168)	80.21	68.03	39.01	47.69	46.89	55.10	66.03	57.57
BERT (1686)	<b>85.38</b>	70.44	52.38	67.53	61.29	49.34	76.09	66.06
GPT3 <sub>Ada</sub> (1686)	<b>85.16</b>	<b>73.88</b>	<b>73.81</b>	75.84	<b>67.42</b>	<b>68.21</b>	<b>76.42</b>	<b>74.39</b>
GPT3.5-In-Context (6)	70.83	50.63	22.47	57.39	43.41	36.30	44.11	46.44
GPT4-In-Context (6)	73.57	61.19	67.60	<b>82.98</b>	55.22	45.81	58.32	63.52

The findings in Table 6.2 and Table 6.3 indicate that GPT3 is more accurate than BERT on both Alien and ALPS data.

Table 6.3 shows that training data size has a large impact on accuracy. Average QWK of GPT3 decreases by 32% (from 69.12 to 46.58) when trained on 10% of the training set. BERT accuracy decreases even more drastically, by 54% (from 64.58 to 29.65). The results indicate that GPT3 is more capable of learning from small data than BERT. In fact, BERT accuracy when trained on 168 examples is comparable to DT, which only uses the number of words in a narrative as a predictive variable.

Our next objective was to compare accuracy of automatic scoring algorithms to human Inter-Rater Reliability (IRR), which can provide an upper bound on the achievable accuracy. GPT3, BERT, and DT were trained using scores provided by the primary human rater. For a subset of 823 narratives from the ALPS test set we had available scores provided by another (secondary) human rater. Both the primary and secondary raters had equal expertise in assessing student narratives. We refer to this set of 823 narratives as the two-rater set, which includes 573 in-sample and 250 out-sample stories (illustrated in Figure 6.4). We reserved the two-rater set for testing. We calculated IRR by treating scores by the primary rater as the ground truth and scores by the secondary rater as the prediction. Then, we calculated QWK by comparing the primary and secondary rater scores. Figure 6.4(A) compares the inter-rater QWK accuracy with QWK accuracy of GPT3, BERT and DT models on the two-rater test set. The horizontal axis distinguishes between different discourse elements and overall average QWK across all discourse elements.

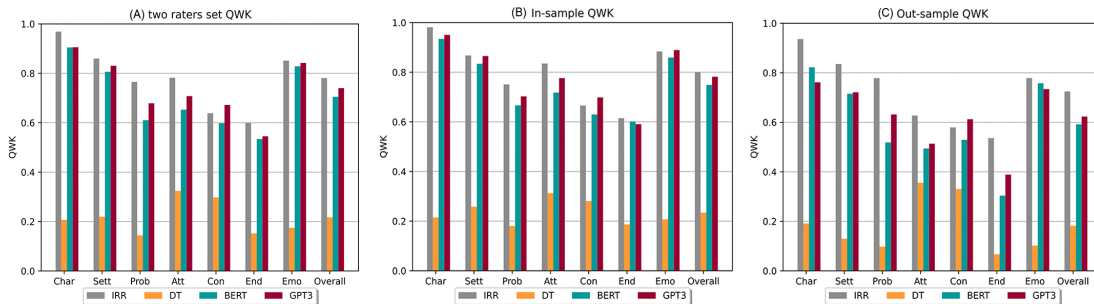


Figure 6.4. IRR comparison of different ML models. Inter Rater Reliability (IRR) compared with ML models on the entire two-rater test set as well as in-sample and out-of-sample portions of the set. The “Overall” bar shows the average QWK scores of ML models across all narrative discourse elements. (A) Annotator agreement on two rater test sets (823 examples). (B) Annotators agreement on in-sample test set (573 examples). (C) Annotators agreement on out-of-sample test set (250 examples).

As seen in Figure 6.4(A), GPT3 had average QWK of 74 and BERT 70. The accuracy is somewhat higher than that reported in Table 6.3, caused by a different distribution of

story types between the full test set and two-rater test set. However, the difference in QWK accuracy between the two models is consistent. The average QWK between the two raters (the IRR bars) is 78. This shows that accuracy of GPT3 is halfway between BERT and human experts and that the accuracy gap between LLMs and humans continues to close. On the individual discourse elements, GPT3 is within the statistical error on Consequence and Emotion.

To evaluate the ability of automatic scoring models to assess types of stories not seen during training, we compared QWK accuracy on in-sample and out-of-sample subset of two-rater test narratives. Figure 6.4(B) and (C) correspond to in-sample and out-of-sample of two-rater narratives. According to the figures, the average QWK scores of GPT3 on in-sample and out-sample data were 78.19 and 62.32, respectively. The average QWK scores of BERT on in-sample and out-of-sample data were 74.8 and 59.16, respectively. Accuracy of both models on out-of-sample data is substantially smaller compared to in-sample data. Only a small fraction of this decrease could be explained by the complexity of assessment of the two groups of story types, because the inter-rater QWK was 79.98 on in-sample and 72.4 on out-of-sample data. The gap between IRR and LLMs GPT3 and BERT is very small on in-sample data, while it is substantial on out-of-sample data.

To select an appropriate GPT3 model that has a good tradeoff between cost and accuracy, we explored different types of GPT models, ranging from the smallest and cheapest Ada to the largest and most expensive Davinci. Due to budget constraints, we conducted this experiment on the ALPS dataset using 10% (168 examples) training data and tested on 10% of the two-rater set (82 examples). We followed the same process explained in Section GPT3 Fine-tuning, to fine-tuned non-Ada GPT3 models Babbage, Curie, and Davinci. We report QWK accuracies of these three larger models in Table 6.4. The results show that there is a relatively small difference in accuracy of the four GPT3 models. Curie was overall the most accurate model and Babbage the least accurate. Davinci was the most expensive model, costing 8.60\$ to fine-tune on 10% of the data. The estimated cost of

fine-tuning Davinci model on the full training set would be 46\$ versus Ada which would be 0.6\$. Given that Ada (0.11\$) is about 8 times cheaper than Curie (0.86\$) and achieves only slightly smaller QWK accuracy, this justifies our choice to use GPT3 Ada in the rest of the experiments.

The results shown so far were about GPT3 model fine-tuned (see Section GPT3 Fine-tuning) on the full training set of 1,686 narratives or on its 10% subset of 168 narratives. As seen from Table 6.3, fine-tuning is data-hungry and requires availability of a large training data set. An alternative to fine-tuning is in-context learning (see Section GPT3 In-cotext). In-context learning does not require expensive fine-tuning, so it is attractive when using more powerful GPT models such as GPT3.5 and GPT4. Another appeal of in-context learning is that it does not need a lot of training data. In our experiment, we used only six rated narratives for the task demonstration, where we randomly selected one narrative from each of the six in-sample story types from the training set. We employed the "chain-of-thought" prompting approach as explained in Section GPT3 In-cotext. This involved providing the model with task instructions along with the six rated narratives. Figure 6.2 illustrates the prompt with one narrative (instead of six) for the demonstration followed by a test narrative that is to be automatically assessed. This method was tested using the most powerful GPT models, GPT3.5 and GPT4, on 10% (82) of the two-rater test narratives.

Table 6.4 compares QWK accuracy of fine-tuning and in-context learning for a range of GPT3, GPT3.5, and GPT4 models on 10% of two-rater test data. Table 6.4 also includes QWK for the IRR and DT model. It is important to highlight that, in the in-context learning setting, we provided only 6 rated narratives, while in fine-tuning, the model was fine-tuned on 168 or 1,686 rated narratives. GPT4 in-context approach was more accurate (63.52 average QWK) than GPT3.5 in-context (46.44 QWK), which is consistent with the existing literature showing that GPT4 is more capable than the older GPT3.5 model (Kozachek, 2023). Importantly, in-context GPT4 accuracy was significantly smaller than the accuracy of GPT3 fine-tuned Ada model (74.39 average QWK) and somewhat smaller than fine-

tuned BERT model (66.06 average QWK) on 1,686 training narratives. Interestingly, fine-tuned GPT3 Ada and BERT on 168 training narratives have lower average QWK accuracy (59.68 and 32.28) than in-context GPT4 that saw only 6 training narratives. This finding indicates that if plentiful training data are available, fine-tuning of less powerful models such as BERT and GPT3 is preferable to in-context learning of GPT3.5 and GPT4. On the other hand, if only a few training stories are available, in-context learning with the more powerful GPT3.5 or GPT4 is preferred.

## 6.4 Discussion

The purpose of this study was to examine the performance of GPT models on automatically assessing narratives. This study extends the work of Jones et al. (2019), who explored automated narrative analysis using BERT Devlin et al. (2018), an immensely successful and popular early LLM.

**Model Performance and Training Data Size:** Our comparative analysis between GPT3 and BERT highlights the GPT3's improved accuracy, particularly on the ALPS data where the training size was larger, which addresses research question Q1-1. Specifically, GPT3's increased accuracy in scoring the Problem, Ending, and Plan/Attempt elements, which are the key plot features, highlights its capabilities in handling complex narrative assessments. The substantial improvements with larger training set compared to a smaller set (GPT3\*\* trained on 10% of the original data), such as a fourfold increase in Problem accuracy and 70% increase in Consequence accuracy, affirms the critical importance of having a lot of training data, which was the focus of Research question Q2-1.

**Inter-rater Reliability Insights:** Comparing accuracy of fine-tuned GPT3 and BERT model with the inter-rater reliability, in the context of question Q1-2, revealed that GPT3 provides a substantial increase in accuracy over the older LLM BERT, particularly for the challenging elements of Ending and Consequence. Comparing with the inter-rater relia-

bility, it can be concluded that GPT3 bridged half of the accuracy gap between BERT and human performance. This suggests that LLMs are getting closer in achieving a human level performance in assessment of discourse elements in our application. It is possible that in cases when the human coders are unavailable or when they are not well trained, GPT3 may already be a feasible assessment tool across various educational contexts and narrative forms.

**Model Generalizability and Performance Variability:** In answering research question Q1-3, we observed a substantial decline in GPT3 and BERT accuracy when assessing story types not seen during training. While GPT3 accuracy on story types seen during training came very close to human performance, the accuracy gap on unseen story types indicates that GPT3 might not be ready for practical use when there is not enough training data for a particular story type. These observations are crucial for informing future model training and development strategies, emphasizing the need for diverse and comprehensive training datasets to enhance model robustness and generalizability.

**Potential of In-context Learning:** When dealing with limited training data, we discovered that in-context learning provides appealing opportunities. Comparison with traditional fine-tuning approaches provides an insightful contrast between the two learning paradigms that could be informative for future applications and development of automated scoring systems. Our results suggest that in situations where acquiring labeled data is difficult, providing a limited number of examples with detailed reasoning for in-context learning of GPT4, one of the most powerful LLMs to date, could result in reasonably accurate narrative assessment. However, if training data are plentiful, our results indicate that fine-tuning smaller and cheaper LLM could result in superior accuracy compared to in-context learning of the most powerful LLMs. It should be added that using more powerful LLMs for narrative assessment is associated with significantly higher monetary costs, which could be a limiting factor in practical applications.

### ***6.4.1 Practical Implications***

Our research indicates that we are reaching a inflection point where automated narrative assessment could become practically useful. The integration of GPT models into the narrative assessment process could offer a way to reduce the burden on educators and allow them to spend more time on instruction and intervention.

Additionally, the adaptability of GPT models facilitates the training on various discourse elements, making it a versatile tool across educational curricula. By leveraging the OpenAI API, users can customize the system to train on specific discourse elements and apply scoring guidelines appropriate for their unique educational contexts.

### ***6.4.2 Limitations and Future Directions***

One potential limitation of the current study is that our results were obtained on assessment of narratives provided by kindergarten to third grade students. Those narratives are typically very short and simple. Thus, it is possible that the obtained results would not be replicated on assessments of longer and more nuanced narratives produced by older students. A limiting factor in repeating our study on narratives from older students would be costs of data collection, as our study indicates that it would be preferable to collect thousands of human rated narratives. Another limitation of our study was that due to the large monetary cost of state-of-the-art LLMs, such as GPT4, some of our results were obtained on subsamples of training and test data. Also, the high monetary cost prevented us from comprehensively exploring prompting strategies for in-context learning and finding good hyperparameters for fine-tuning. Thus, it is possible that we could have reached higher accuracy in our experiments if we were able to better optimize different aspects of our experimental procedure.

There are many possible directions for future research. For example, it would be beneficial to pursue an end-to-end integration of Automatic Speech Recognition (ASR) tools

with Large Language Models (LLMs). This integration could streamline the process of generating high-quality narrative transcripts for training. As another example, fine-tuning LLMs on training data that include textual justifications for the assessed scores presents a promising avenue for future exploration. We hypothesize that including justifications could enhance the model's comprehension of the assessment. Since collection of training data enriched with justifications is expensive, it opens interesting questions about costs and best approaches for preparing training data and using such data for training automated narrative assessment models.

## **6.5 Conclusions**

In this study, we explored the potential of GPT, the state-of-the-art class of LLMs, as an automated scoring system for assessing discourse elements in narrative language samples. Fine-tuned GPT3 outperformed fine-tuned BERT, the older-generation LLM model, showing impressive performance across all narrative elements. We demonstrated the critical importance of having plentiful training data for fine-tuning. Fine-tuning smaller LLMs on large training data can be better than fine tuning larger LLMs on less data. In situations where training data are extremely limited, in-context learning of the most powerful models such as GPT4 can achieve impressive assessment accuracy. Our results indicate that accuracy of fine-tuned LLMs substantially decreases on story types unseen during training. On story types seen during training, fine-tuned GPT3 approaches and in some cases matches accuracy of human experts. Overall, our results imply that current state-of-the-art LLMs such as the GPT class of models might already be a feasible solution for assessment in specific application scenarios, while in other scenarios there are still significant obstacles to their deployment. Further research is needed to better understand the scenarios where LLMs are applicable for automatic assessment of student work and to comprehensively evaluate benefits and pitfalls of deploying such systems in practice.

## CHAPTER 7

### CONCLUSION

This dissertation presents a series of innovative methodologies to address the challenges of document classification in low-resource environments, focusing on healthcare and education domains. The research tackles the critical issues of annotation efficiency, model training with limited labeled data, and automated evaluation, providing practical solutions that significantly enhance the performance and applicability of machine learning models in these fields.

In the healthcare domain, the developed visualization approach for rapid labeling of clinical notes has shown to dramatically reduce the cognitive load on annotators, resulting in faster and more efficient data labeling. By clustering similar sentences and highlighting key features, this tool accelerates the annotation process, enabling the collection of larger volumes of high-quality labeled data. This, in turn, improves the training and accuracy of machine learning models used for tasks such as smoking status extraction.

The research further addresses the problem of training accurate classifiers with limited labeled data through the MERIT framework. By leveraging shortest dependency paths (SDP) and specific distance thresholds for label propagation, this method effectively augments labeled datasets, enhancing model performance. The iterative algorithm developed to learn automatic thresholds for label propagation extends this approach, demonstrating substantial improvements in various learning scenarios, including semi-supervised, supervised, and in-context learning.

In the education domain, the dissertation proposes a novel approach to automated narrative scoring using large language models (LLMs). This methodology accurately captures the scoring patterns of teachers, offering a scalable and consistent alternative to manual evaluation. By reducing the subjectivity and resource demands associated with manual

scoring, this approach provides a reliable tool for assessing narratives generated by school-aged children.

Overall, the experimental results validate the effectiveness of the proposed methodologies in improving annotation speed, data utilization, and model accuracy. The contributions of this dissertation offers scalable and practical solutions for critical tasks in healthcare and education.

### ***Limitations & Future Research***

A limitation of the study in Chapter 3 is the lack of evaluation of how annotation quality impacts final model accuracy. While the proposed display reduced annotation time, the quality of the annotations was not assessed.

Additionally, the within-subject design, where annotators used both the proposed and baseline displays, may have introduced learning or fatigue effects, influencing performance. Future research could address this by employing a between-subject design to eliminate carryover effects. However, this would require careful group balancing and a larger sample size for robust comparisons.

In Chapter 6, the study could benefit from involving additional annotators to achieve more reliable inter-rater reliability (IRR), particularly for challenging elements. A limitation is the lack of clear instructions on how annotators were trained for scoring. Furthermore, the use of one expert and one non-expert annotator in the Alien dataset introduces variability that could affect consistency. Future research would benefit from employing annotators with comparable expertise and including more than two raters to improve agreement reliability.

While these limitations point to specific methodological improvements, the thesis also opens up broader opportunities for future research. Beyond addressing these challenges, there are several promising directions for advancing the techniques and applications introduced in this work. These directions aim to extend the impact of this research by exploring

innovative methods and tackling new problems. The primary future directions can be summarized into three objectives.

**Objective 1: SDP-Based Applications in Rapid Labeling** While this research leveraged SDP-based label propagation methods to tune classifiers in low-resource settings, an intriguing direction for future work would be to extend the application of SDP-based representations to the annotation process itself, particularly in sentence clustering tasks. Using SDP representations as a basis for clustering could facilitate more complex annotation tasks, such as Relation Extraction (RE), by grouping sentences with similar contextual and syntactic structures. Future research could investigate the effectiveness of SDP-based clustering in improving annotation efficiency and consistency, potentially reducing cognitive load for annotators and increasing the quality of labeled datasets.

**Objective 2: LLM-Based Relation Extraction with Self-Supervised SDP Pre-Training** Building on the success of SDP-based label propagation, future research could explore self-supervised learning approaches to pre-train SDP representations on large, unannotated biomedical corpora. This approach would involve designing self-supervised objectives that help the model capture dependency-based relationships within biomedical text. For instance, a Masked Dependency Path Prediction task—where specific tokens or segments within dependency paths are masked, and the model is trained to predict them—could encourage the model to learn syntactic and semantic patterns essential for relation extraction. Additionally, Dependency Path-Aware Attention mechanisms could be developed to enhance the model’s focus on tokens within the SDP, ensuring that the embeddings capture key relational cues. Such attention layers could assign greater weight to syntactically significant tokens, producing refined and relation-specific SDP-based representations.

**Objective 3: GPT-Based Classifier with Reasoning for Educational Applications** Extending the current application of narrative scoring, future research could focus on

integrating reasoning explanations alongside scoring, making the feedback more informative for educational purposes. While collecting ground-truth reasoning data can be resource-intensive, one approach could involve generating synthetic explanations aligned with ground-truth scores, and using these generated explanations to fine-tune a GPT-based model. By training the model on paired scores and explanations, this approach could create a classifier capable not only of assigning accurate scores but also of providing reasoning behind each score. This added layer of feedback could improve the interpretability and educational value of automated scoring systems for student narratives.

## BIBLIOGRAPHY

- Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., & Salakoski, T. (2008), “All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning,” *BMC bioinformatics*, 9, 1–12.
- Almubark, N. M., Silva-Maceda, G., Foster, M. E., & Spencer, T. D. (2023), “Indices of Narrative Language Associated with Disability,” *Children*, 10, 1815.
- Bairoch, A. & Apweiler, R. (1997), “The SWISS-PROT protein sequence data bank and its supplement TrEMBL,” *Nucleic acids research*, 25, 31–36.
- Baldini Soares, L., FitzGerald, N., Ling, J., & Kwiatkowski, T. (2019), “Matching the Blanks: Distributional Similarity for Relation Learning,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2895–2905, Florence, Italy, Association for Computational Linguistics.
- Beltagy, I., Lo, K., & Cohan, A. (2019), “SciBERT: A pretrained language model for scientific text,” *arXiv preprint arXiv:1903.10676*.
- Bishop, D. V. M. & Edmundson, A. (1987), “Language-impaired 4-year-olds: Distinguishing transient from persistent impairment,” *Journal of speech and hearing disorders*, 52, 156–173.
- Boudjellal, N., Zhang, H., Khan, A., & Ahmad, A. (2020), “Biomedical relation extraction using distant supervision,” *Scientific Programming*, 2020, 8893749.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020), “Language models are few-shot learners,” *Advances in neural information processing systems*, 33, 1877–1901.
- Bunescu, R., Ge, R., Kate, R. J., Marcotte, E. M., Mooney, R. J., Ramani, A. K., & Wong, Y. W. (2005), “Comparative experiments on learning information extractors for proteins and their interactions,” *Artificial intelligence in medicine*, 33, 139–155.
- Catts, H. W., Fey, M. E., Tomblin, J. B., & Zhang, X. (2002), “A longitudinal investigation of reading outcomes in children with language impairments,” *Journal of Speech, Language, and Hearing Research*.
- Chen, J., Ji, D., Tan, C. L., & Niu, Z.-Y. (2006), “Relation extraction using label propagation based semi-supervised learning,” in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 129–136.
- Church, K. W. (2017), “Word2Vec,” *Natural Language Engineering*, 23, 155–162.

- Collins, M. & Duffy, N. (2001), “Convolution kernels for natural language,” *Advances in neural information processing systems*, 14.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018), “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019), “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)*, pp. 4171–4186, Minneapolis, Minnesota, Association for Computational Linguistics.
- Dey, R. & Salem, F. M. (2017), “Gate-variants of gated recurrent unit (GRU) neural networks,” in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pp. 1597–1600, IEEE.
- Dickinson, D. K. & McCabe, A. (2001), “Bringing it all together: The multiple origins, skills, and environmental supports of early literacy,” *Learning Disabilities Research & Practice*, 16, 186–202.
- Dikli, S. (2006), “An overview of automated scoring of essays,” *The Journal of Technology, Learning and Assessment*, 5.
- Dinh, T., Zeng, Y., Zhang, R., Lin, Z., Rajput, S., Gira, M., Sohn, J.-y., Papailiopoulos, D., & Lee, K. (2022), “Lift: Language-interfaced fine-tuning for non-language machine learning tasks,” *arXiv preprint arXiv:2206.06565*.
- Enayati, S., Yang, Z., Lu, B., & Vucetic, S. (2021), “A visualization approach for rapid labeling of clinical notes for smoking status extraction,” in *Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances*, pp. 24–30.
- Enayati, S., Yang, Z., Vucetic, S., & Spencer, T. D. (2024), “Automated Narrative Scoring Using Large Language Models (dataset),” .
- Erkan, G., Ozgur, A., & Radev, D. (2007), “Semi-supervised classification for extracting protein interaction sentences using dependency parsing,” in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 228–237.
- Fang, M., Yin, J., & Tao, D. (2014), “Active Learning for Crowdsourcing Using Knowledge Transfer,” *Proceedings of the AAAI Conference on Artificial Intelligence*, 28.
- Fernandez, N., Ghosh, A., Liu, N., Wang, Z., Choffin, B., Baraniuk, R., & Lan, A. (2022), “Automated scoring for reading comprehension via in-context bert tuning,” in *Artificial Intelligence in Education: 23rd International Conference, AIED 2022, Durham, UK, July 27–31, 2022, Proceedings, Part I*, pp. 691–697, Springer.

- for Best Practices, N. G. A. C. & of Chief State School Officers, C. (2010), *Common Core State Standards for English Language Arts*, National Governors Association Center for Best Practices, Council of Chief State School Officers, Washington, DC.
- Fox, C., Jones, S., Gillam, S. L., Israelsen-Augenstein, M., Schwartz, S., & Gillam, R. B. (2022), “Automated Progress-Monitoring for Literate Language Use in Narrative Assessment (LLUNA),” *Frontiers in Psychology*, 13.
- Gillam, R. B. & Pearson, N. A. (2004a), *Test of narrative language*, Pro-ed.
- Gillam, R. B. & Pearson, N. A. (2004b), *TNL: Test of narrative language*, Pro-ed Austin, TX.
- Gillam, S. L., Gillam, R. B., Fargo, J. D., Olszewski, A., & Segura, H. (2017), “Monitoring indicators of scholarly language: A progress-monitoring instrument for measuring narrative discourse skills,” *Communication Disorders Quarterly*, 38, 96–106.
- Gillon, G., McNeill, B., Scott, A., Gath, M., & Westerveld, M. (2023), “Retelling stories: The validity of an online oral narrative task,” *Child Language Teaching and Therapy*, p. 02656590231155861.
- Good, B. M. & Su, A. I. (2013), “Crowdsourcing for bioinformatics,” *Bioinformatics*, 29, 1925–1933.
- Greenhalgh, K. S. & Strong, C. J. (2001), “Literate language features in spoken narratives of children with typical language and children with language impairments,” *Language, Speech, and Hearing Services in Schools*.
- Gupta, P., Rajaram, S., Schütze, H., & Runkler, T. (2019), “Neural relation extraction within and across sentence boundaries,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 6513–6520.
- Han, J. M., Xu, T., Polu, S., Neelakantan, A., & Radford, A. (2021), “Contrastive finetuning of generative language models for informal premise selection,” in *6th Conference on Artificial Intelligence and Theorem Proving*.
- Hanneke, S. (2009), *Theoretical foundations of active learning*, Carnegie Mellon University.
- Hassanali, K.-n., Liu, Y., & Solorio, T. (2012), “Evaluating NLP Features for Automatic Prediction of Language Impairment Using Child Speech Transcripts.” in *INTER-SPEECH*, pp. 1339–1342.
- Hassantabar, S., Dai, X., & Jha, N. K. (2019), “STEERAGE: Synthesis of neural networks using architecture search and grow-and-prune methods,” *arXiv preprint arXiv:1912.05831*.

- Hassantabar, S., Wang, Z., & Jha, N. K. (2021), “SCANN: Synthesis of compact and accurate neural networks,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41, 3012–3025.
- Hegde, H., Shimpi, N., Glurich, I., & Acharya, A. (2018), “Tobacco use status from clinical notes using Natural Language Processing and rule based algorithm,” *Technology and Health Care*, 26, 445–456.
- Herrero-Zazo, M., Segura-Bedmar, I., Martínez, P., & Declerck, T. (2013), “The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions,” *Journal of biomedical informatics*, 46, 914–920.
- Hsu, C.-J. & Thompson, C. K. (2018), “Manual versus automated narrative analysis of agrammatic production patterns: The Northwestern Narrative Language Analysis and Computerized Language Analysis,” *Journal of Speech, Language, and Hearing Research*, 61, 373–385.
- Hu, X., Zhang, C., Yang, Y., Li, X., Lin, L., Wen, L., & Yu, P. S. (2021), “Gradient Imitation Reinforcement Learning for Low Resource Relation Extraction,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.
- Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., & Mark, R. G. (2016), “MIMIC-III, a freely accessible critical care database,” *Scientific data*, 3, 160035.
- Jones, S., Fox, C., Gillam, S., & Gillam, R. B. (2019), “An exploration of automated narrative analysis via machine learning,” *Plos one*, 14, e0224634.
- Katic, T., Pavlovski, M., Sekulic, D., & Vucetic, S. (2021), “Learning Semi-Structured Representations of Radiology Reports,” *arXiv preprint arXiv:2112.10746*.
- Khalpada, P. & Garg, S. (2021), “Simple Automated Narrative Generator (SANG),” in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0909–0915, IEEE.
- Kim, B. & Pardo, B. (2018), “A human-in-the-loop system for sound event detection and annotation,” *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8, 1–23.
- Kim, Y.-S., Al Otaiba, S., & Wanzek, J. (2015), “Kindergarten predictors of third grade writing,” *Learning and Individual Differences*, 37, 27–37.
- Klie, J.-C., Eckart de Castilho, R., & Gurevych, I. (2020), “From Zero to Hero: Human-In-The-Loop Entity Linking in Low Resource Domains,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6982–6993, Online, Association for Computational Linguistics.

- Köhler, G., Bode-Böger, S., Busse, R., Hoopmann, M., Welte, T., & Böger, R. (2000), “Drug-drug interactions in medical patients: effects of in-hospital treatment and relation to multiple drug use.” *International journal of clinical pharmacology and therapeutics*, 38, 504–513.
- Krallinger, M., Rabal, O., Akhondi, S. A., Pérez, M. P., Santamaría, J., Rodríguez, G. P., Tsatsaronis, G., Intxaurre, A., López, J. A., Nandal, U., et al. (2017), “Overview of the BioCreative VI chemical-protein interaction Track,” in *Proceedings of the sixth BioCreative challenge evaluation workshop*, vol. 1, pp. 141–146.
- Krasakis, A. M., Kanoulas, E., & Tsatsaronis, G. (2018), “Semi-supervised ensemble learning with weak supervision for biomedical relationship extraction,” in *Automated Knowledge Base Construction (AKBC)*.
- Kuboyama, T., Hirata, K., Kashima, H., Aoki-Kinoshita, K. F., & Yasuda, H. (2007), “A spectrum tree kernel,” *Information and Media Technologies*, 2, 292–299.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015), “Deep learning,” *nature*, 521, 436–444.
- Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020), “BioBERT: a pre-trained biomedical language representation model for biomedical text mining,” *Bioinformatics*, 36, 1234–1240.
- Li, M., Takamura, H., & Ananiadou, S. (2020), “A Neural Model for Aggregating Coreference Annotation in Crowdsourcing,” in *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 5760–5773, Barcelona, Spain (Online), International Committee on Computational Linguistics.
- Li, Z., Yang, Z., Shen, C., Xu, J., Zhang, Y., & Xu, H. (2019), “Integrating shortest dependency path and sentence sequence into a deep learning framework for relation extraction in clinical text,” *BMC medical informatics and decision making*, 19, 1–8.
- Lim, S., Jatowt, A., Färber, M., & Yoshikawa, M. (2020), “Annotating and Analyzing Biased Sentences in News Articles using Crowdsourcing,” in *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 1478–1484, Marseille, France, European Language Resources Association.
- Lin, H., Yan, J., Qu, M., & Ren, X. (2019), “Learning dual retrieval module for semi-supervised relation extraction,” in *The World Wide Web Conference*, pp. 1073–1083.
- Liu, H., Hunter, L., Kešelj, V., & Verspoor, K. (2013), “Approximate subgraph matching-based literature mining for biomedical events and relations,” *PloS one*, 8, e60954.
- Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., & Chen, W. (2021), “What Makes Good In-Context Examples for GPT-3?” *arXiv preprint arXiv:2101.06804*.

- Liu, L. (2018), “Heterogeneous Supervision for Relation Extraction,” in *Mining Structures of Factual Knowledge from Text: An Effort-Light Approach*, pp. 119–127, Springer.
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023), “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing,” *ACM Computing Surveys*, 55, 1–35.
- Lucy, L. & Bamman, D. (2021), “Gender and representation bias in GPT-3 generated stories,” in *Proceedings of the Third Workshop on Narrative Understanding*, pp. 48–55.
- MacWhinney, B. & Snow, C. (1985), “The child language data exchange system,” *Journal of child language*, 12, 271–295.
- Malekzadeh, M., Hajibabae, P., Heidari, M., Zad, S., Uzuner, O., & Jones, J. H. (2021), “Review of graph neural network in text classification,” in *2021 IEEE 12th annual ubiquitous computing, electronics & mobile communication conference (UEMCON)*, pp. 0084–0091, IEEE.
- Meng, Y., Shen, J., Zhang, C., & Han, J. (2018), “Weakly-supervised neural text classification,” in *proceedings of the 27th ACM International Conference on information and knowledge management*, pp. 983–992.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013), “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*.
- Miller, J. F., Andriacchi, K., Nockerts, A., Westerveld, M. F., & Gillon, G. (2016), *Assessing language production using SALT software: A clinician’s guide to language sample analysis*, SALT Software, LLC Middleton, WI.
- Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009), “Distant supervision for relation extraction without labeled data,” in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 1003–1011.
- Mizumoto, A. & Eguchi, M. (2023a), “Exploring the potential of using an AI language model for automated essay scoring,” *Research Methods in Applied Linguistics*, 2, 100050.
- Mizumoto, A. & Eguchi, M. (2023b), “Exploring the potential of using an AI language model for automated essay scoring,” *Research Methods in Applied Linguistics*, 2, 100050.
- Moschitti, A. (2006), “Efficient convolution kernels for dependency and constituent syntactic trees,” in *European Conference on Machine Learning*, pp. 318–329, Springer.
- Needleman, S. B. & Wunsch, C. D. (1970), “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *Journal of molecular biology*, 48, 443–453.

- Neumann, M., King, D., Beltagy, I., & Ammar, W. (2019), “ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing,” in *Proceedings of the 18th BioNLP Workshop and Shared Task*, pp. 319–327, Florence, Italy, Association for Computational Linguistics.
- Nguyen, H. T. & Smeulders, A. (2004), “Active learning using pre-clustering,” in *Proceedings of the twenty-first international conference on Machine learning*, p. 79.
- Ouali, Y., Hudelot, C., & Tami, M. (2020), “An overview of deep semi-supervised learning,” *arXiv preprint arXiv:2006.05278*.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022), “Training language models to follow instructions with human feedback,” *arXiv preprint arXiv:2203.02155*.
- Palaga, P. (2009), “Extracting relations from biomedical texts using syntactic information,” *Mémoire de DEA, Technische Universität Berlin*, 138.
- Palmer, E. L., Hassanpour, S., Higgins, J., Doherty, J. A., & Onega, T. (2019), “Building a tobacco user registry by extracting multiple smoking behaviors from clinical notes,” *BMC medical informatics and decision making*, 19, 1–10.
- Papanikolaou, Y. & Pierleoni, A. (2020), “Dare: Data augmented relation extraction with gpt-2,” *arXiv preprint arXiv:2004.13845*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011), “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, 12, 2825–2830.
- Petersen, D. B. & Spencer, T. D. (2012), “The narrative language measures: Tools for language screening, progress monitoring, and intervention planning,” *Perspectives on Language Learning and Education*, 19, 119–129.
- Peterson, C. & McCabe, A. (1983), “Three ways of looking at a child’s narrative: A psycholinguistic analysis,” *NY: Plenum*.
- Qian, K., Popa, L., & Li, Y. (2020), “An Intuitive User Interface for Human-in-the-loop Entity Name Parsing and Entity Variant Generation,” in *Proceedings of (DaSH@KDD)*, Association for Computing Machinery.
- Qu, M., Ren, X., Zhang, Y., & Han, J. (2018), “Weakly-supervised relation extraction by pattern-enhanced embedding learning,” in *Proceedings of the 2018 World Wide Web Conference*, pp. 1257–1266.
- Rajendran, S. & Topaloglu, U. (2020), “Extracting Smoking Status from Electronic Health Records Using NLP and Deep Learning,” *AMIA Summits on Translational Science Proceedings*, 2020, 507.

- Ramesh, D. & Sanampudi, S. K. (2022), “An automated essay scoring systems: a systematic literature review,” *Artificial Intelligence Review*, 55, 2495–2527.
- Ranade, P., Dey, S., Joshi, A., & Finin, T. (2022), “Computational Understanding of Narratives: A Survey,” *IEEE Access*, 10, 101575–101594.
- Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., & Ré, C. (2020), “Snorkel: rapid training data creation with weak supervision,” *The VLDB Journal*, 29, 709–730.
- Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D., & Ré, C. (2016), “Data programming: Creating large training sets, quickly,” *Advances in neural information processing systems*, 29.
- Ren, W., Li, Y., Su, H., Kartchner, D., Mitchell, C., & Zhang, C. (2020), “Denoising multi-source weak supervision for neural text classification,” *arXiv preprint arXiv:2010.04582*.
- Rosenberg, C., Hebert, M., & Schneiderman, H. (2005), “Semi-supervised self-training of object detection models,” .
- Rubin, O., Herzig, J., & Berant, J. (2021), “Learning to retrieve prompts for in-context learning,” *arXiv preprint arXiv:2112.08633*.
- Sajjadi, M., Javanmardi, M., & Tasdizen, T. (2016), “Regularization with stochastic transformations and perturbations for deep semi-supervised learning,” *Advances in neural information processing systems*, 29.
- Sammut, C. & Webb, G. I. (2011), *Encyclopedia of machine learning*, Springer Science & Business Media.
- Scott, C. M. & Windsor, J. (2000), “General language performance measures in spoken and written narrative and expository discourse of school-age children with language learning disabilities,” *Journal of speech, language, and hearing research*, 43, 324–339.
- Segura-Bedmar, I., Martínez Fernández, P., & Herrero Zazo, M. (2013), “Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013),” Association for Computational Linguistics.
- Settles, B. (2011), “Closing the Loop: Fast, Interactive Semi-Supervised Annotation With Queries on Features and Instances,” in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 1467–1478, Edinburgh, Scotland, UK., Association for Computational Linguistics.
- Seung, H. S., Opper, M., & Sompolinsky, H. (1992), “Query by Committee,” in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92*, p. 287–294, New York, NY, USA, Association for Computing Machinery.
- Silva, O. (1999), “Guide to Narrative Language: Procedures for Assessment,” *Estudios filológicos*, pp. 204–205.

- Smola, A. & Vishwanathan, S. (2002), “Fast kernels for string and tree matching,” *Advances in neural information processing systems*, 15.
- Snow, C. E., Porche, M. V., Tabors, P. O., & Harris, S. R. (2007), *Is literacy enough? Pathways to academic success for adolescents.*, Paul H. Brookes Publishing Co.
- Solmaz, G., Cirillo, F., Maresca, F., & Kumar, A. G. A. (2022), “Label Augmentation with Reinforced Labeling for Weak Supervision,” *arXiv preprint arXiv:2204.06436*.
- Somasundaran, S., Lee, C., Chodorow, M., & Wang, X. (2015), “Automated scoring of picture-based story narration,” in *Proceedings of the tenth workshop on innovative use of NLP for building educational applications*, pp. 42–48.
- Spencer, T. & Staff (2023), “Academic Language of Primary Students (ALPS),” Available at: <https://nyu.databrary.org/volume/1632> (Accessed: 27 February 2024).
- Stein, N. L. & Glenn, C. G. (1975), “An Analysis of Story Comprehension in Elementary School Children: A Test of a Schema.”
- Sun, X., Li, X., Li, J., Wu, F., Guo, S., Zhang, T., & Wang, G. (2023), “Text classification via large language models,” *arXiv preprint arXiv:2305.08377*.
- Susanti, M. N. I., Ramadhan, A., & Warnars, H. L. H. S. (2023), “Automatic essay exam scoring system: a systematic literature review,” *Procedia Computer Science*, 216, 531–538.
- Tashu, T. M., Maurya, C. K., & Horvath, T. (2022), “Deep Learning Architecture for Automatic Essay Scoring,” *arXiv preprint arXiv:2206.08232*.
- Tikk, D., Thomas, P., Palaga, P., Hakenberg, J., & Leser, U. (2010), “A comprehensive benchmark of kernel methods to extract protein–protein interactions from literature,” *PLoS computational biology*, 6, e1000837.
- Von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S., & Bork, P. (2002), “Comparative assessment of large-scale data sets of protein–protein interactions,” *Nature*, 417, 399–403.
- Wang, L., Ruan, X., Yang, P., & Liu, H. (2016), “Comparison of three information sources for smoking information in electronic health records,” *Cancer informatics*, 15, CIN–S40604.
- Wang, M., Min, F., Zhang, Z.-H., & Wu, Y.-X. (2017), “Active learning through density clustering,” *Expert systems with applications*, 85, 305–317.
- Wang, R., Chen, D., & Kwong, S. (2013), “Fuzzy-rough-set-based active learning,” *IEEE Transactions on Fuzzy Systems*, 22, 1699–1704.

- Wang, R., Chow, C.-Y., & Kwong, S. (2015), “Ambiguity-based multiclass active learning,” *IEEE Transactions on Fuzzy Systems*, 24, 242–248.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022), “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in neural information processing systems*, 35, 24824–24837.
- Wei, Q., Ji, Z., Si, Y., Du, J., Wang, J., Tiryaki, F., Wu, S., Tao, C., Roberts, K., & Xu, H. (2019), “Relation extraction from clinical narratives using pre-trained language models,” in *AMIA annual symposium proceedings*, vol. 2019, p. 1236, American Medical Informatics Association.
- Wilkinson, G. R. (2005), “Drug metabolism and variability among patients in drug response,” *New England Journal of Medicine*, 352, 2211–2221.
- Wishart, D. S., Knox, C., Guo, A. C., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., & Woolsey, J. (2006), “DrugBank: a comprehensive resource for in silico drug discovery and exploration,” *Nucleic acids research*, 34, D668–D672.
- Zhang, S., He, L., Dragut, E., & Vucetic, S. (2019), “How to Invest My Time: Lessons from Human-in-the-Loop Entity Extraction,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, p. 2305–2313, New York, NY, USA, Association for Computing Machinery.
- Zhang, Y., Zhong, V., Chen, D., Angeli, G., & Manning, C. D. (2017), “Position-aware attention and supervised data improve slot filling,” in *Conference on Empirical Methods in Natural Language Processing*.
- Zhang, Y., Zheng, W., Lin, H., Wang, J., Yang, Z., & Dumontier, M. (2018), “Drug–drug interaction extraction via hierarchical RNNs on sequence and shortest dependency paths,” *Bioinformatics*, 34, 828–835.
- Zhou, D., Bousquet, O., Lal, T., Weston, J., & Schölkopf, B. (2003), “Learning with local and global consistency,” *Advances in neural information processing systems*, 16.
- Zhou, W., Lin, H., Lin, B. Y., Wang, Z., Du, J., Neves, L., & Ren, X. (2020), “Nero: A neural rule grounding framework for label-efficient relation extraction,” in *Proceedings of The Web Conference 2020*, pp. 2166–2176.
- Zhou, W., Huang, K., Ma, T., & Huang, J. (2021), “Document-level relation extraction with adaptive thresholding and localized context pooling,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 14612–14620.
- Zhu, X. & Ghahramani, Z. (2002), “Learning from labeled and unlabeled data with label propagation,” .